CRANFIELD UNIVERSITY

SCHOOL OF INDUSTRIAL MANUFACTURING SCIENCE

Ph.D. THESIS

Academic Year 1997-98

ADE IDOWU

DYNAMIC METROLOGY OF ERROR MOTIONS IN PRECISION SPINDLES USING OPTICAL METROLOGY

Supervisor: Dr A E Gee

October 1998

CRANFIELD UNIVERSITY

SCHOOL OF INDUSTRIAL MANUFACTURING SCIENCE

Ph.D. THESIS

Academic Year 1997-98

ADE IDOWU

DYNAMIC METROLOGY OF ERROR MOTIONS IN PRECISION SPINDLES USING OPTICAL METROLOGY

Supervisor: Dr A E Gee

October 1998

This thesis is submitted in fulfilment for the Degree of Doctor of Philosophy

ABSTRACT

Knowledge of the accuracies of air bearing spindles in the sub-micrometre to nanometre range is required for the design, commissioning and operation of ultra-precise machine tools, measurement systems and other machines employing high precision rotational motion.

In order to verify the dynamic performance of a spindle, measurement is required of its error motions in the unwanted five degrees of freedom (one axial, two tilts and two radial motions). Presentation of these error motions (eg in the form of polar charts) can then be used to provide critical spindle metrology data including *total*, *asynchronous* and *average* error motion *rosette* profiles and their average and peak values.

This thesis describes a metrology system based on optical interferometry for measuring such unwanted error motions in three degrees of freedom involving motion along the spindle axis (axial rectilinear displacement and tilts about orthogonal axes), incurred with rotation of a precision air spindle over its specified speed-range. The system is not sensitive to orthoaxial translations which may be measured using alternative methods.

Possible alternative techniques for measuring any of the degrees of freedom include an array of proximity sensors, (one for each translational degree of freedom and a further one for each of the other rotational degrees of freedom), to measure the run-out of an artefact. Proximity sensors based upon capacitive or optical fibre back-scatter techniques each offer the required single degree-of-freedom non-contacting capability and bandwidth.

In the current work, a Fizeau interferometer is used to monitor the motion of the spindle of a vertical axis ultra-precision facing machine using a test-artefact. This is a mirror with less than one fringe departure from planarity from which interferogram fringe-patterns are captured, digitised and analysed synchronously as the spindle rotates. The issue of the prediction of the dynamic form and motions of the observed interferogram arises and the earlier theory is extended to optimise the set-up, including provision of automatic servo-alignment of the optical axis with the axis of the spindle.

Measurement interferogram data is sampled at selected angular incremental positions of spindle-rotation and image processing techniques used to filter the fringe pattern, enabling measurement of spindle tilt and axial displacement. Issues of sampling with respect to the anticipated spatial angular frequency of the spindle run-out are considered with respect to the speed/frequency capability of data-acquisition and processing arrangements. Essentially, with a spindle rotating at typical machining speeds of 300-3000 rev/min, for consistent error motions, the resolution of an error plot is principally a function of observational time. It is foreseen that the system will be applicable in research and production-support in ultra-precision machining production processes and in rotational metrology.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 INTRODUCTION TO CHAPTER

There is a general need in the field of precision engineering for advancing the level of accuracy of precision machine tools (eg such as diamond turning machines, grinding machines) and also for surface, roundness, and form measuring systems (eg Talysurf*, Talyrond*, Talycenta* measuring instruments and CMMs - Co-ordinate Measuring Machines). A critical underlying and developing area of work involves studying the accuracy of precision and ultra-precision spindles for these machines/instruments.

In this chapter a brief description of the application of air bearings and the need for spindle metrology in precision engineering is given. A literature review of the developments in air bearing performance is also provided.

1.2 APPLICATIONS OF AIR BEARING SPINDLES IN PRECISION ENGINEERING

According to Collier and Wilson (1982), a spindle bearing can be defined as a machine element which facilitates motion between a shaft (rotor) and a fixed part (stator), or a part moving relative to a shaft. Generally bearings can be categorised as:

- metal-to-metal (rolling contact)
- hydrodynamic
- hydrostatic
- air (gas) bearings

According to Powell (1970), a gas bearing can be defined as two accurately machined surfaces separated by a thin film of gas and arranged so that any tendency to change clearance between the surfaces is resisted by a change in the gas film. There are broadly four types of air bearings, which can be classified as:

- aerodynamic
- squeeze film
- aerostatic
- hybrid (combined aerodynamic and aerostatic) bearings

In a aerodynamic bearing, pressure is generated when one surface moves relative to another so that the lubricant (air) is dragged into a convergence between the surfaces (Powell 1970) as shown schematically in figure 1.1a.

---

* Taylor Hobson Co, New Star Road, Leicester

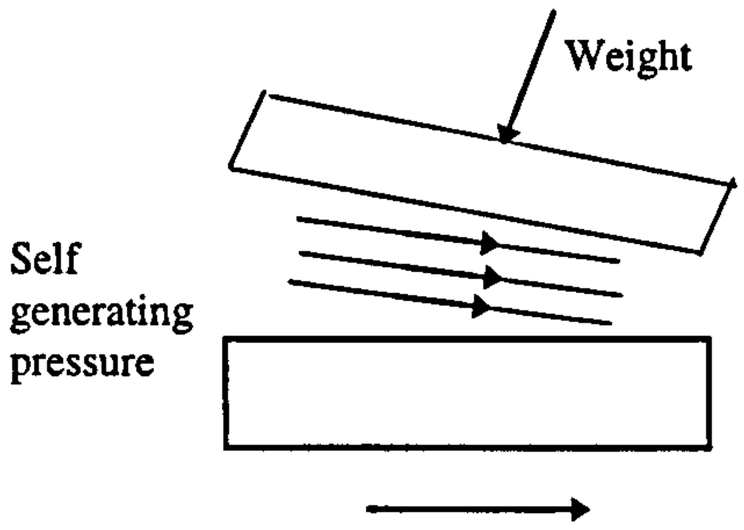


Figure 1.1a: Aerodynamic

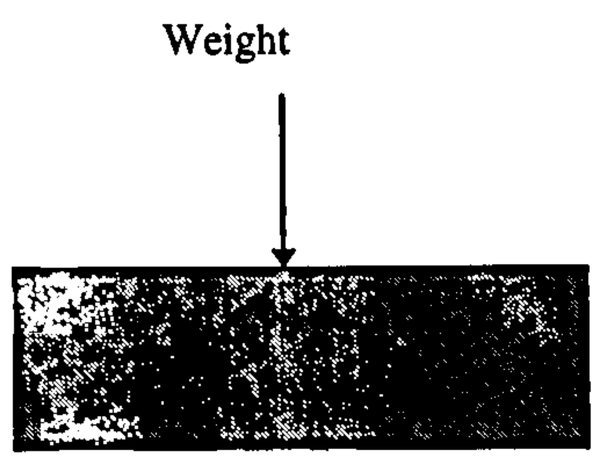


Figure 1.1b: Squeeze film

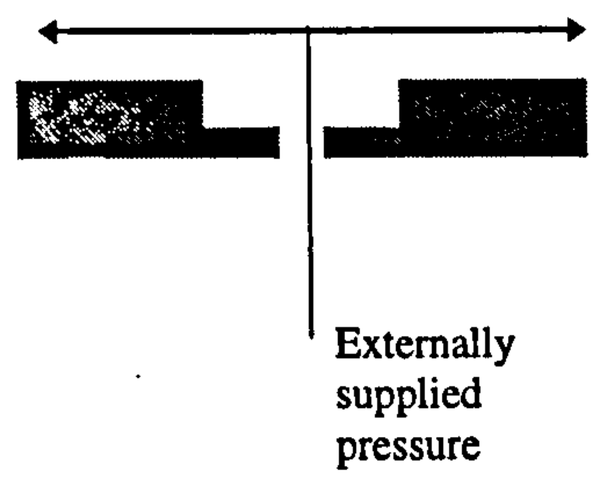


Figure 1.1c: Aerostatic

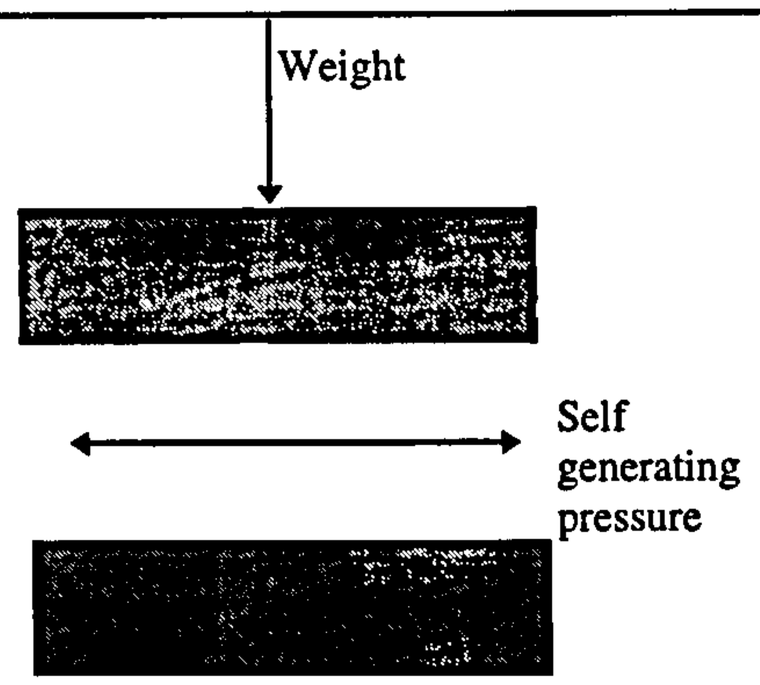


Figure 1.1a, 1.1b and 1.1c: Types of air bearings after Powell (1970)

On the other hand, in the squeeze film bearing, pressure is generated by the oscillatory motion of one of the bearing surfaces (Powell, 1970) as shown schematically in figure 1.1b. Both the aerodynamic and squeeze film bearing operate without any external gas supply and hence they are self acting bearings.

On the contrary, aerostatic bearings (see figure 1.1c) require an external supply of pressure. The gas from the external source is fed into the bearing clearance with the aid of restrictors (which are usually grooves or slots or feed holes machined on to the bearing surfaces). An aerostatic bearing spindle was used in this work (see section 2.2 for more details).

There are a variety of bearing geometries; these include:

- cylindrical journal
- circular thrust
- annular thrust
- conical
- spherical

Air bearings are increasingly being used in a number of applications which traditionally incorporated other types of bearings. This is due to their accurate motion and low friction. Typical engineering applications of air bearings include, machine tools, precision measuring systems, gas circulators (used in the nuclear industry), turbo-compressors (used in the power industry), auxiliary equipment (used in cryogenic systems), and gyroscopes for military applications.

Other applications of air bearings include products/mechanisms which range from surgical tools (such as drills and cleaning equipment) used in dentistry to Winchester disk drives in personal computers Pan (1990).

Typical advantages of using air bearings in precision engineering are:

- Minimal friction and heat generation due to the low viscosity of air (Wunsch 1965), hence the bearings are very tolerant of small changes in clearance caused by viscous heating (Slocum 1992, pp580 - 625).

- Good rotational accuracy of air spindle due to the averaging action of the pressurised air film around the circumference of spindle rotor (Wunsch 1965).

- Very low bearing film thickness can be maintained and this increases the stiffness of the machine, since the thickness is inversely proportional to bearing stiffness.

- Air bearings have zero static friction (Slocum 1992, pp580 - 625) and very low dynamic friction at moderate spindle speeds (1000 - 3000 rpm).

- Due to its low friction and thus low wear rate, the bearing life span is very high.

- Aerostatic bearings are generally self-cleaning.

- Air bearing grinding machines have longer wheel life, and do not require frequent wheel dressing (Wunsch, 1965), because their spindles rotate accurately.

1.3 HISTORICAL DEVELOPMENTS OF PRECISION AIR SPINDLES

According to Pan (1990) the first discovery of air lubrication was made by French scientist Gustav Adolph Hirn in 1854. Hirn made this discovery while experimenting on the bearing effects of water and air lubricants in a *friction balance* instrument.

By 1886 Englishman Osborne Reynolds developed a comprehensive theory of fluid film lubrication between two surfaces. This work was used to explain experimental results of Tower's (1885) study of an oil-lubricated bearing. Reynolds (1886) developed differential equations which describe the relationship between pressure distribution, film thickness and fluid viscosity - assuming incompressible, isothermal and iso-viscous conditions.

In 1896 American scientist, Albert Kingsbury rediscovered the phenomenon of air lubrication in a *compression device*, and subsequently went on to study an air-lubricated journal bearing (Kingsbury, 1897). English scientist Harrison (1913) developed solutions for infinitely long gas-lubricated slider and cylindrical air bearings. He also pioneered the technique of numerical computation of air bearing problems (Pan, 1990). In which his solution of air-lubricated journal bearings of infinite length was evaluated by (manual) numerical integration using the Runge method.

Widespread industrial application of air bearing spindles was hardly in evidence at the beginning of this century apart in high speed textile machinery. Limited applications of air spindles in machine tools was witnessed during this period, such as its application in high speed grinding machines by Sawtell (1909).

It was not until the early 1950s that industrial applications of air spindles increased. According to Powell (1970), the two specialised applications that stimulated the early development of air spindle bearings during this era, were in the area of high precision gyroscopes for use in inertial navigation systems, and in the development of gas circulators used in the nuclear industry.

Today, precision spindles are used in a number of areas which include precision instruments, machine tools, dental drills, industrial refrigerators and liquefiers, high-speed electric motors, computer disk drives and other forms of fabrication equipment required to operate at very high rotational accuracy. Figure 1.2 shows a schematic representation of some applications of precision spindles.

1.4 NEED OF SPINDLE METROLOGY IN PRECISION ENGINEERING

McKeown (1983) defined precision engineering as the grouping of engineering and scientific skills and techniques in response to the increasing application of metrology to manufacturing. Precision engineering is therefore concerned with the manufacture of materials and components, the development of advanced manufacturing technologies and the design/manufacture of precision machine tools and measurement systems, and their control systems. Spindle metrology in precision engineering is used to achieve the required (very high) accuracy of machine tools and their end products, and precision instruments.

There are number of areas in precision engineering where there is a need for spindle metrology, some of these are described in the following sub-sections:

1.4.1 FABRICATION OF I-T AND AERO-ENGINE COMPONENTS

The spindle is one of the principal components of a machine tool and its performance is judged by its ability to produce a workpiece of required tolerance. This performance is dependent on the static and dynamic behaviour of all components that form the *structural loop* [the structural loop is part of the machine tool that is prone to vibration, it includes the tool-holder, guide-ways, bed, spindle, work-holder and the workpiece itself Bryan & Vanherck (1975)]. The spindle is normally the most flexible component within the structural loop, and in order for optimal machine performance the stiffness of the spindle component should be infinitely high.

Generally, bearing run-out limits the performance of machine tool spindles and the accuracy of their end products. For instance, fabrication of lenses, semiconductors and fibre optics require fine surface machining (within sub-micrometre to nanometre scale), and spurious spindle motion would affect the surface form of the machined component.

Over the past decade, single-point diamond machining has become a growing area of precision engineering for the manufacturing of computer magnetic memory discs substrates, convex mirrors for high output carbon dioxide laser resonators, spherical bearing surfaces in beryllium and copper, infra-red lenses, scanners for printers and X-ray mirror substrates. Very high surface finishes (i.e. of about 3 to 4 nm peak-to-valley) and geometric accuracy can be maintained under carefully controlled machining conditions and by the application of high stiffness aerostatic bearings McKeown(1986).

Free abrasive machining processes such as grinding, polishing and lapping have been used traditionally for the manufacture of lenses. Today there is a growing area of the manufacture of advanced technology components such as magnetic disk systems and optical discs by abrasive machining processes. These processes requires spindle wheels with low error motions and compliance.

Major improvements in thermal efficiency are potentially obtainable from the application of ceramic materials in the manufacture of critical components of aero-engines due to the material's refractory properties. This technological breakthrough requires *ductile regime* machining of ceramics at a stable critical depth of cut (normally of the order of about one micrometre according to Puttick et al 1989), requiring spindles to run to sub-micrometre levels.

1.4.2 ACCURACY OF MEASURING INSTRUMENTS (FOR MECHANICAL PARAMETERS)

Air spindles are increasingly used in measuring instruments to achieve very high rotational accuracy. Typical applications of air spindles in measurement instruments include: roundness measuring machines, rotary CMMs, dynamometers; others include gyro-instruments, high-speed ball-bearing test rigs, wind-tunnel balances and guided missile instrumentation (Grassam & Powell, 1964).

A roundness measuring instrument is used for measuring the out-of-roundness, concentricity and ovality of artefacts such as shafts, cylinder bores and spherical artefacts. It typically employs a vertical shaft supported in an air journal bearing. Some very accurate roundness instruments incorporate an air-thrust bearing on the top and bottom flanges of the bearing rotor, this is used to support the weight of the shaft and artefact under test. The instrument may use a fixed or rotating *stylus* (depending on its configuration) for measuring the radial variations in diameter of the test-piece. A typical commercial roundness measuring machine is the Taylor Hobson Talyrond, which has measurement resolutions that range from sub-micrometre to nanometre levels.

In roundness measuring instruments such as the Talyrond, spindle run-out errors introduce ambiguity to readings in the measurements taken during a particular roundness test. These could also cause alignment problems between the axes of the spindle rotation and the test artefact. This necessitates the requirement for very precise spindle metrology.

Rotary CMMs are instruments used for obtaining dimensional metrology information of rotational test-pieces. Unlike other CMM configurations, the rotary CMM incorporates a vertical spindle turn-table on which the test-piece is placed and a movable probe. According to Ramesh (1990) the volumetric accuracy of CMMs, which is the maximum error between any two points in a specified swept volume of a CMM depends on: geometrical and displacement (linear and rotary) accuracy. Where the latter requires accurate rotational motion during the design and application/calibration stage of the instrument. Guidelines for accurate application/calibration of CMM errors are documented in the American standard ANSI/ASME (1990) and BSI British standards (1987-89).

In the case of dynamometers (which are instruments used for measuring motor torque), in order to obtain an accurate measurement of torque, it is essential to mount the motor in journal bearings which have frictional torque values relatively less than the measured (motor) torque (Grassam & Powell, 1964). This requires a journal air spindle with minimum friction and accurate rotation.

SUB-MICROMETRE TO NANOMETRE SPINDLE ACCURACY AND LOW SPEED

MACHINE TOOLS:

- Diamond turning M/Cs
- Grinding M/Cs
- Microelectronic fabrication M/Cs
- Other types of precision M/Cs

INSTRUMENTS:

- Roundness measuring M/Cs
- CMMs
- Dynamometers

PRECISION

SPINDLES

ELECTRO-MECHANICAL DRIVES:

- Magnetic disks drives
- Optical disks drives

OTHER APPLICATIONS:

- High speed drills for medical applications
- Bearing machine test rigs
- Disk brakes for automotive applications
- Gyro systems and gas turbines
- Gas circulators for nuclear industry
- Spinning disk sprays for process industry

SUB-MILLIMETRE TO SUB-MICROMETRE SPINDLE ACCURACY AND MEDIUM TO HIGH SPEED APPLICATIONS

Figure 1.2: Application of precision spindles and measurement

1.4.3 ELECTRO-MECHANICAL DEVICES

Precision air spindles are widely used in electro-mechanical devices such as magnetic and optical disc drives. There has been a gradual increase in the application of magnetic storage devices (Talke 1987), this is due to:

- the invention and implementation of *disc file* (this is a mechanism which incorporates a magnetic head supported on a thin film of air, in close proximity to a rotating disc)
- availability of magnetic *read/write heads* which can be positioned in close proximity to a flexible magnetic tape without excessive wear

In order to improve the functionality of a magnetic hard-disc, it is important that the disc has a very high storage density (Talke, 1987). Over the last 30 years there has been a linear increase in the storage density (as shown in figure 1.3), this has been due to improvements in the read/write head, enhancement in the signal processing electronics and to a large degree improvements in the precision mechanical design of the recording devices. Future improvement in magnetic disc storage (McKeown, 1986) will depend on reducing the:

- head-to-disc spacing (which is typically between 0.2 and 0.3 μm),
- read/write gap length
- disc magnetic coating thickness

These attributes will require spindles which operate with nanometric maximum non-repeatable run-out (Talke, 1987).

Another growing area of disc storage is in the application of optical discs, theses are currently used for storing/retrieving audio and video data, and other forms of digital-storage applications. Increased speed and storage capacity of optical discs will depend on accurate positioning of the disc/read-write head and on the accuracy of rotation of the disc about its axis.

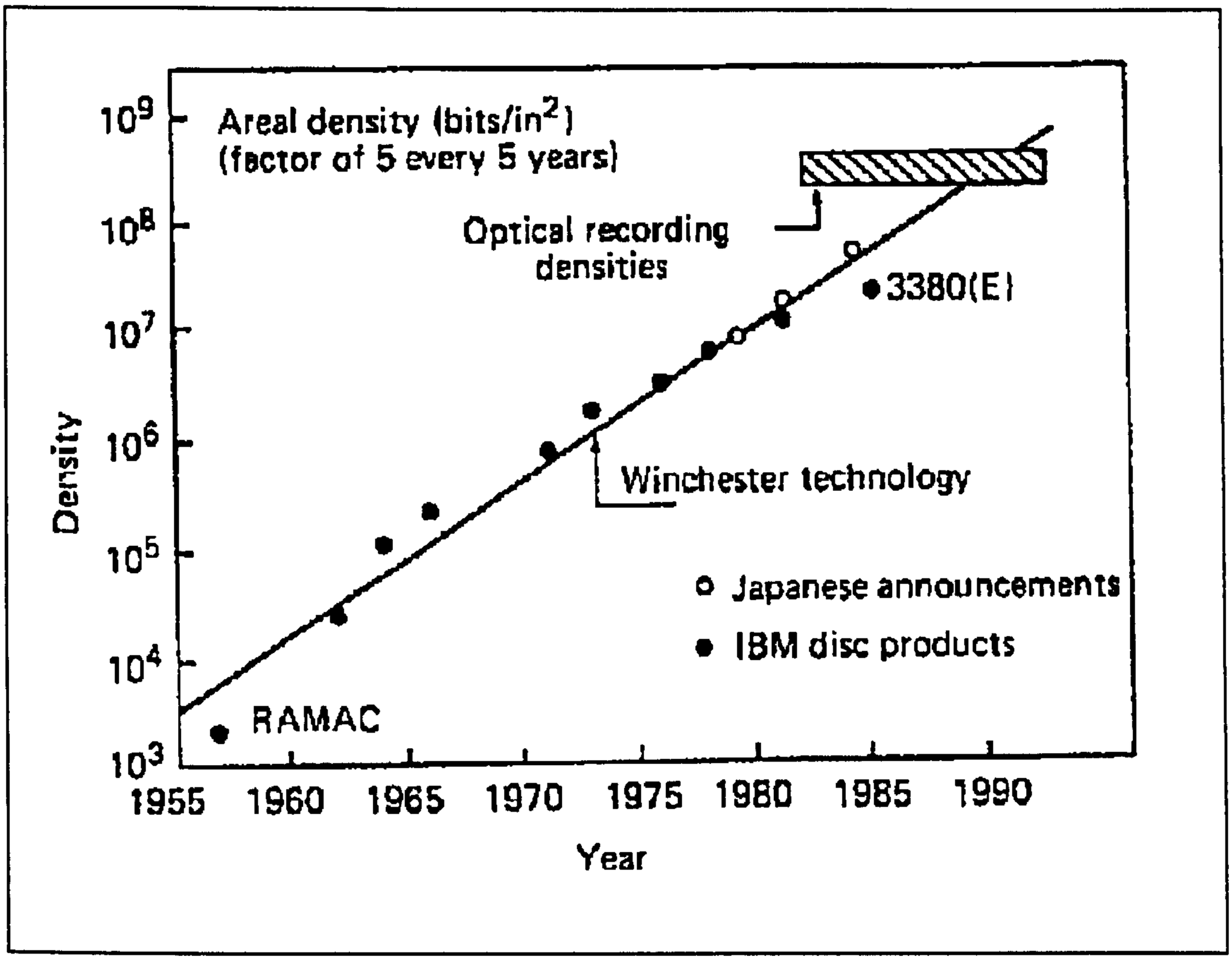


Figure 1.3: Storage density of magnetic discs (after Talke, 1987)

1.4.4 TESTING OF BRAKE DISCS FOR AUTOMOTIVE APPLICATIONS

Knowledge of rotor run-out and thickness of disk brakes is required when designing and developing safe braking systems for automobiles. Lion precision developed a capacitive spindle error analyser for the non-contact measurement of brake disk error motions and thickness variations; see Lion (1992) for more details.

1.5 LITERATURE REVIEW OF DEVELOPMENTS IN AIR SPINDLE PERFORMANCE

Many theoretical and experimental research studies have been undertaken to characterise the performance of air spindles. The following sub-sections describe some of these studies:

1.5.1 THEORETICAL ANALYSIS OF AIR BEARINGS

Theoretical research into spindle accuracy has mainly involved the application of analytical, numerical and computer simulation methods for analysing the variations of bearing fluid film clearance, pressure distribution and thermal flux, and bearing dynamic constants (stiffness and damping coefficients).

Before the advent of computational methods, fluid lubrication problems were solved analytically. One of the earliest analytical evaluations of fluid lubrication between two surfaces was by Reynolds (1886), in which the so-called Reynolds' equation was developed. During the pioneering days of air lubrication development only a few researchers such as Harrison (1913) solved Reynolds equation numerically. This was because the solution required sophisticated (digital) computation which did not exist until the late 1950s.

As a result of the complexities encountered in solving the analytical solution of Reynolds' equation for particular gas bearings problems, numerical solutions or computer simulation methods based on the modified Reynolds' equation have been used. For instance Castelli & Pirvics (1968) reviewed different types of numerical solution methods for time dependent and time independent gas bearing problems.

Currently computer simulation and modelling techniques of gas bearing systems is a growing area. Typical bearing analysis software can categorised as: *computational fluid dynamic (CFD)* software programs based on finite-element or finite-volume or difference equations methods (Anderson, 1995) and, dynamic simulation packages software programs that provides a visualisation of the bearing dynamic effects.

A number of finite-element and finite-volume modelling techniques have been developed over the last decade. Most of theses techniques evaluate the pressure gradient within the bearing clearance and use this information to calculate the variation in of the bearing gap at specified angular positions.

Hendricks (1988) developed a finite-element software package used for modelling steady state gas bearings. This development incorporated most of the details of bearing and was essentially intended to be a generic design tool for characterising the performance of air bearings. This software was used to analyse the tilt error motions of a thrust bearing of a laser scanner as it revolved at 30,000 rpm. Results obtained from this analysis indicated that at a clearance of 10 μm the bearing exhibited a tilt error motion of 200 μrads.

Oshumi et al (1984) developed a theoretical and experimental frequency response for an aerostatic bearing in which plots of *amplitude ratio* (bearing clearance variation divided by the varying applied load) and *relative phase* as functions of bearing *rotational frequency* and compressibility were generated. The response was used to investigate how to make a bearing absorb as much vibratory energy as possible (ie to reduce its compliance to ideally zero) and establish a new design criterion for the bearing.

Yabe (1994) numerically analysed the run-out characteristics of externally pressurised gas journal bearings, giving special attention to the influence between the run-out of the bearing rotor and its machining error. This numerical analysis involved the application of the divergence formulation technique to calculate the exact solution of the bearing clearance pressure distribution, taking in to consideration the influence of rotor and bearing pocket circularity errors. This work indicated that spindle run-out is governed by rotor out-of-roundness and bearing pocket size deviation (ie which also affects spindle stiffness)

Dong-chul et al (1994) used the *direct numerical* method to evaluate stiffness and damping coefficients of an aerostatic bearing, This approach allowed the radial run-out of bearing rotor to be solved numerically. Subsequent experimental evaluation of the bearing run-out was also carried out using capacitive probes, and the maximum difference between the measured and calculated radial error motion was less than 5%.

Iordanoff et al (1995) investigated the effect of misalignment and inertia on high-speed thrust bearings using a theoretical approach. This approach involved a numerical solution of Reynolds' equation of flow for the bearing in its three degrees of freedom (ie representing its orthogonal tilts and the axial error motions). This resulted in the evaluation of bearing dynamic coefficients, and subsequently in the modal analysis of the bearing which was used to analyse the 3 degrees of freedom modes of vibration (error motions).

The main draw-back of some theoretical techniques used for modelling spindle accuracy, is that they give an approximation of the true performance of a bearing. This is usually due to:

- the simple geometry employed in the model which does not take in to consideration the complex profile of the bearing gap or rotor out-of-roundness

- discretisation error of the model mesh applied to the areas and volumes considered (Anderson, 1995).

Accurate evaluation of most bearings usually require an experimental analysis which can be validated using a theoretical approach. For instance, Huges et al (1996) analysed a gas lubricated hydrodynamic thrust bearing using both theoretical and experimental techniques. The theoretical technique involved a numerical solution of Reynolds' equation and the *equation of state*, using a finite-volume modelling program. The experimental technique on the other hand, involved the measurement of bearing gap (bearing run-out) using eddy-current sensors and the evaluation of bearing pressure and temperature profiles using an array of pressure and thermocouple transducers respectively. This development was used for investigating different pivot pad bearing geometry and the effects of pressure distribution and bearing gap variations during isothermal and non-isothermal conditions. Results from the theoretical approach and its corresponding experimental work were observed to correlate very well. The measured bearing gap fluctuations at speeds grater-than 10,000 rpm was 6 μm.

Yabe and Ishida (1991) evaluated the influence of forced vibrations on axial error motions of an aerostatic thrust bearing using both theoretical and experimental techniques. The experimental technique employed an eddy-current sensors for measuring the axial error displacement. While the theoretical approach used a *perturbation solution* for evaluating the static and dynamic characteristics of the bearing under varying conditions of: applied load (W), pocket size (k) and feed parameter (Γ). Frequency response obtained for the theoretical and experimental evaluation of spindle axial error (d_z) at spindle speeds of 0 to 20,000 rpm confirmed that at large applied loads, resonance frequency is inversely proportional to d_z. The evaluation (theoretical and experimental) of spindle axial error motion in this work, varied from 0.1 - 8.0 μm for applied axial loads (normalised) of 0.075 - 0.175.

1.5.2 EXPERIMENTAL ANALYSIS OF AIR BEARINGS

One of the earliest examples of work involving the experimental measurement of bearing performance was that by Schlesinger (1938), who developed a method of testing the accuracy of rotation of machine tool spindles using a dial indicator to measure radial run-out. This particular method had shortcomings from a precision engineering point of view; it was very difficult to distinguish between simple eccentricity in the workpiece and in the bearing, the method could not indicate the course of the deviation vector during rotation, and the indicator could only be used to determine the displacements (error motions) in one axis at a time.

As a result of these limitations, Tlusty (1959) pioneered the technique of using two mutually perpendicular sensitive deviation pickups (such as inductive or capacitive sensors) to measure simultaneously the position of a cylindrical mandrel attached to the spindle under test. In this work the signals from two sensors were connected respectively to the horizontal and vertical axes of an oscilloscope respectively. His work made it possible for the first time to visualise spindle run-out errors dynamically on an oscilloscope.

The work of Bryan et al (1967) represented a much more in-depth analysis of spindle accuracy. It involved the measurement of the spindle radial error motion using capacitively generated *Lassajous* figures. In this work two capacitive probes were used to generate a polar chart *base circle*, and a third probe was used to measure the radial error of a spherical master attached to a spindle. These signals were used to generate the required polar chart, indicating spindle radial error motion.

Nowadays capacitive sensing of spindle error motions is a widely used technique in precision engineering. For instance Chapman (1986), developed a commercial spindle analyser which incorporated an array of capacitive sensors used to measure (to nanometric resolution) the 5 degree of freedom error motions of a rotating spindle. Similarly, Martin (1995), has developed a spindle analysis instrument which incorporates software that enables error motions to be analysed and plotted on a computer as ; *asynchronous, average, inner/outer* and *total error motions.*

Bryan & Vanherck (1975) developed definitions of spindle metrology terms. This work was modernised during the development of the ANSI/ASME American Standard (1985).

A number of optical techniques have been developed in the past to analyse spindle run-out, these are usually *interferometric*, *moiré* or *triangulation* techniques. Optical interferometric measurement of spindle performance has a considerable advantage due to its *full field* measurement capability, non-invasiveness, low noise and high resolution typically between λ\100 to λ\1000 (depending on the accuracy of the interferogram analysis technique).

One of the earliest examples of an interferometrically based measurement technique applied to gas bearings was carried out by Stone (1921). This involved the application of optical interferometry to measure the pressure profile and clearance variation of the air film in a laboratory tilting-pad thrust bearing. The bearing consists of a glass disc (rotor) and a quartz block (stator). The block was positioned above the surface of the (rotating) disc with the aid of a spring mechanism. Interference fringes were formed in the air gap between the two optical surfaces of the bearing, and the number of fringes observed was used to quantify the variation of bearing clearance. According to Pan (1990) this development was to become one of the most widely used techniques in contemporary studies of gas lubrication.

Gee et al (1988) at Cranfield devised a technique of interferometrically monitoring the spindle and workpiece on an ultra-precision single-point diamond facing machine. The main requirement of this work was to detect the error motions of the spindle (based on a known surface form), and to evaluate surface/shape during rotation. This work was further developed by Gee (1993) to include analysis of interferogram dynamics of a rotating artefact.

In a development of this by Triandis et al (1991), a Fizeau interferometer was set-up above and essentially coaxial with the vertical spindle axis of a research diamond turning machine. This enabled the axial, tilt and form errors of the mounted plane workpiece (which ideally should be normal to the spindle axis) to be monitored to sub-micrometre accuracy (to about one tenth of a fringe) and corrected in-process with the aid of a micro-tilt stage mechanism attached to the spindle. However during this research the measurement of the spindle performance was carried out at fixed positions of the spindle. The work involved in this project (ie work explained in this thesis) is intended to demonstrate the possibility of measuring the spindle run-out dynamically (ie while the spindle is rotating).

Kim and Park (1994) developed an optical moiré method for measuring radial error motions of an air spindle. This used two concentric gratings of fine pitch configured in a way such that the radial error motions were detected by analysing the interferometric fringes generated by the gratings. This method was immune to mechanical and electrical disturbances, since neither a master cylinder nor electrical gauge was required. It was capable of measuring the spindle run-out to a resolution of 1 μm.

Recent work carried out by Risse and Guyenot (1996) used a laser triangulation technique to quantify the amount of spindle *wobble* to arc seconds. This method worked on the basis of reflecting a laser beam off one of 18 polygonal mirror facets on a (glass) aerostatic spindle bearing arrangement, and then further reflecting the beam off a mirror onto a high resolution CCD detector (i.e. the CCD line had 2048 pixels with a pixel size of 13 μm). This set-up enabled the spindle orthogonal tilts to be quantified, based on the position of the laser spot measured by the CCD camera. It was used to measure the tilt error motion of 0.96 μrad as the spindle revolved at 3,000 rpm.

Other research related to spindle accuracy involves both the measurement of spindle run-out and in-process correction of error motions. Mizuno (1993) developed a means of measuring the tilt and axial error components of a diamond turning machine tool spindle by the application of optical fibre displacement sensors and then corrected these errors by using a piezo-electric micro-tilt stage mechanism with the aid of PID and repetitive control algorithms. This set-up was able to detect spindle axial error motion (before and after an in-process error correction) of 100 nm and 10 nm respectively.

Chen and Yang (1989), developed dynamic compensation technology for the spindle error motion of a precise lathe. This set-up used four capacitive sensors to measure the axial and orthogonal radial error motions (ie with aid of a master cylinder clamped to the spindle face plate). Signals obtained from the capacitive sensors were then amplified and sent to the on-board tool holder compensation set-up to control in-situ, the influence of the spindle out-of roundness during the machining process. Dynamic tests of the lathe showed that the roundness error (due to spindle run-out) reduced from 1.0 - 1.7 μm to 0.4 - 0.8 μm as a result of the in-process spindle error motion correction.

On the other hand Horikawa et al (1992) also developed an active air journal bearing which utilised capacitive sensors to detect the spindle radial error motion and non-contact actuators (i.e. moveable air pads driven by piezo-electric actuators) to control and regulate the bearing. This work showed that the active journal bearing had almost infinite static stiffness and an absolute rotational accuracy better than 21 nm as the spindle revolved at 750 rpm.

CHAPTER 2: DESCRIPTION OF AEROSTATIC BEARING SPINDLE

2.1 INTRODUCTION

A research diamond turning machine incorporating a Blockhead type 4R aerostatic bearing, manufactured by Professional Instruments Company (PI)* was used in this project. Aerostatic spindle bearings are used in many ultra-precision machine tools and measuring instruments, due to their very high rotational accuracy. In this chapter a general description of the PI bearing design and its operational issues will be presented.

2.2 APPLICATION OF PI BEARING - RESEARCH DIAMOND TURNING MACHINE

The research diamond turning machine used in this project described by Puttick et al (1989), was designed originally to investigate single point diamond machining of brittle materials, under fixed depth of cut. The machine consists of an aerostatic spindle used to rotate the workpiece, and a hinged tool arm arrangement. This arrangement guides the travel of the tool as it is fed across the workpiece in a plane orthogonal to the spindle axis (Duduch 1993). Figure 2.1 depicts the schematic layout of the machine and figure 2.2 shows a photograph of the machine. The machine has subsequently been modified (Gee et al, 1988) to accommodate a vacuum chuck and an interferometer.

The principle mechanical elements of the machine are:

- **Main structure**: supports the aerostatic spindle assembly, it also carries the weight of the hinged tool post and the interferometer as shown in figures 2.1 and 2.2. This part of the machine structure is reasonably stiff, and any induced vibrations from the floor are isolated by the application of four isolators.

- **Air spindle**: See section 2.3 for details of the aerostatic spindle used in this machine.

- **Vacuum chuck**: The machine incorporates a vacuum chuck arrangement for holding the workpiece on the face of the spindle. The vacuum chuck was also designed to be used as a manually adjusted tilt stage mechanism. Figure 2.3 shows the schematic of the vacuum chuck (Duduch 1993).

---

* Professional Instruments Co., 4601 Highway 7, Minneapolis 55416, USA

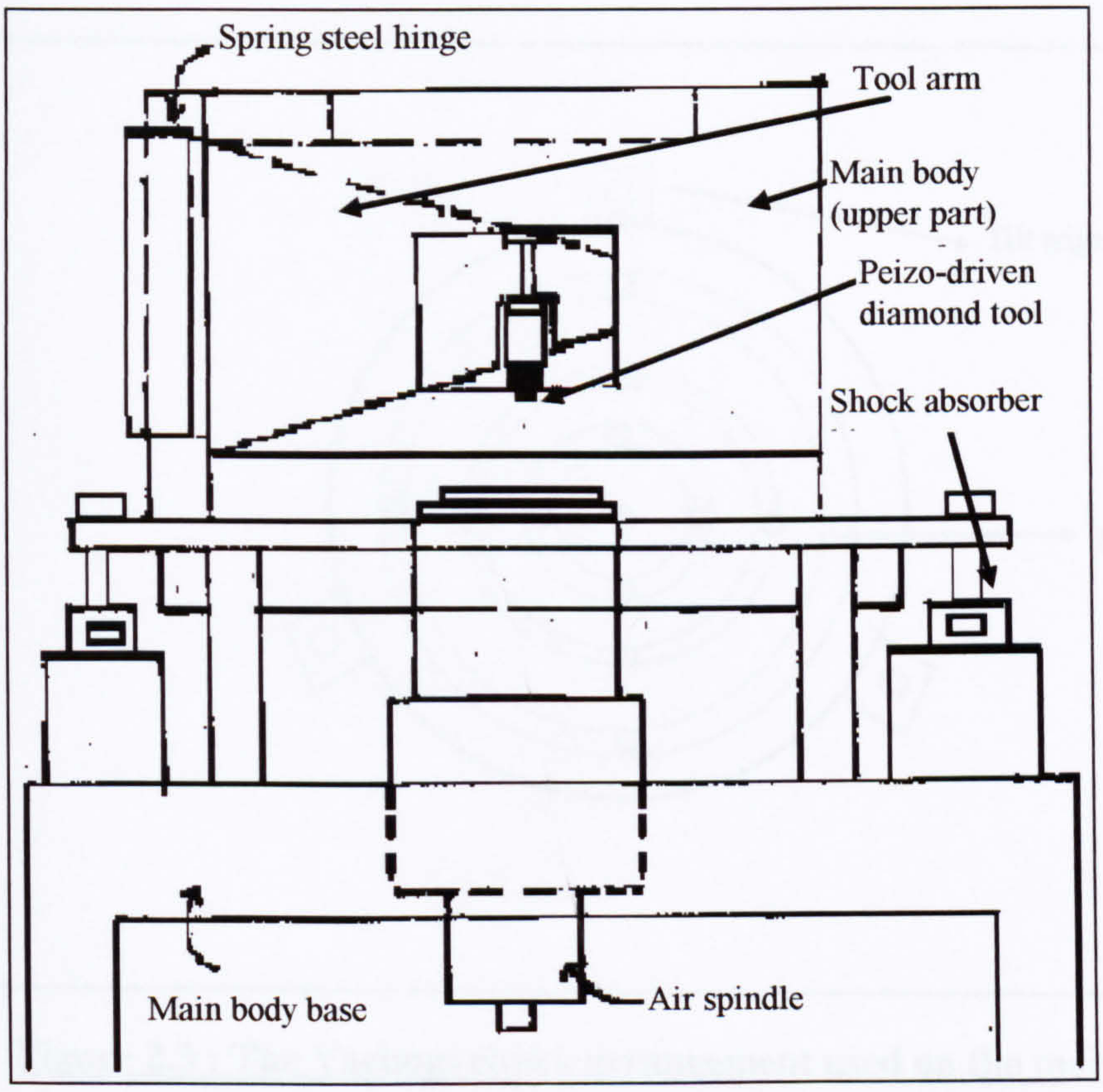


Figure 2.1 Schematic layout of the research turning machine

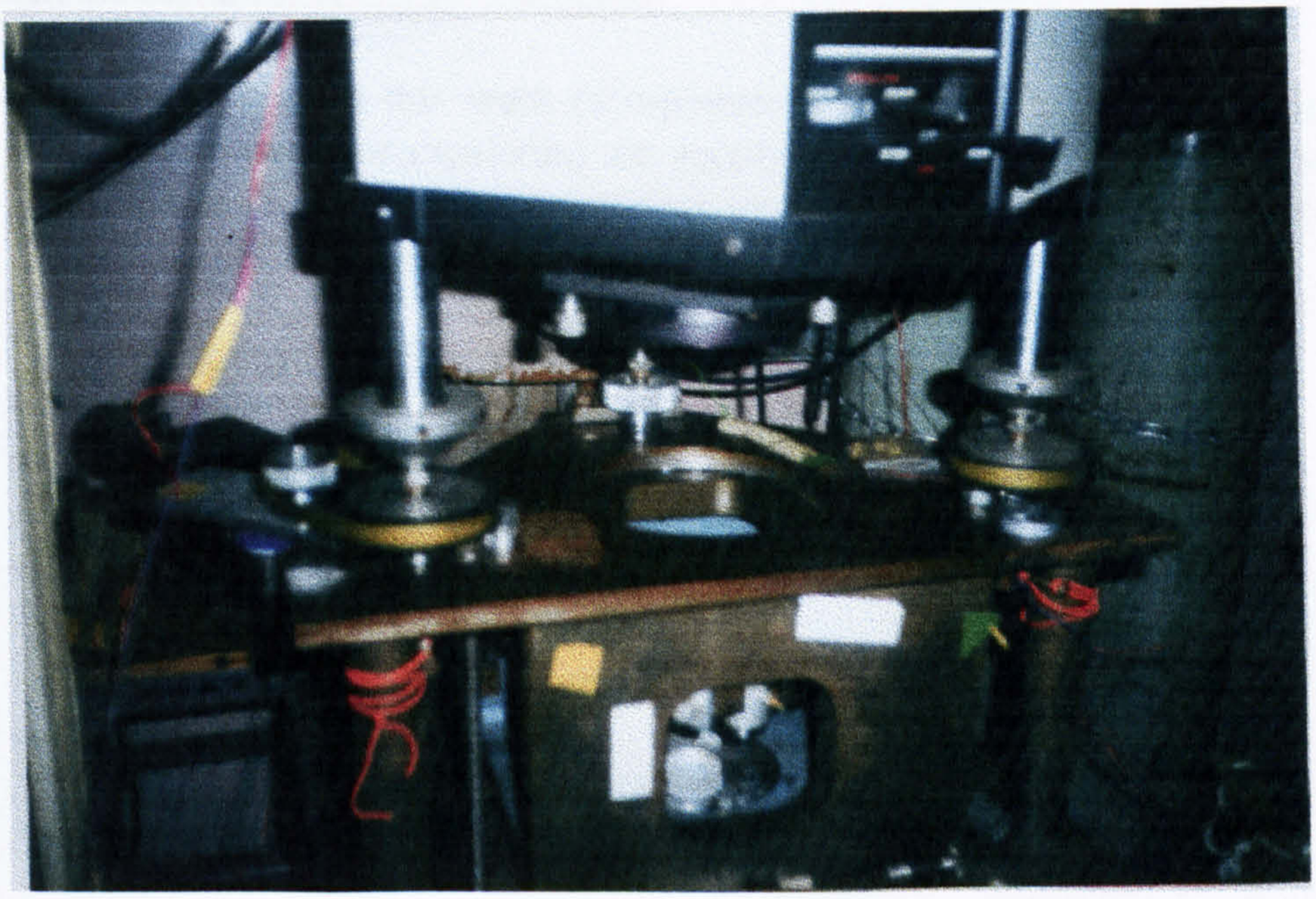


Figure 2.2: Photograph of the research turning machine

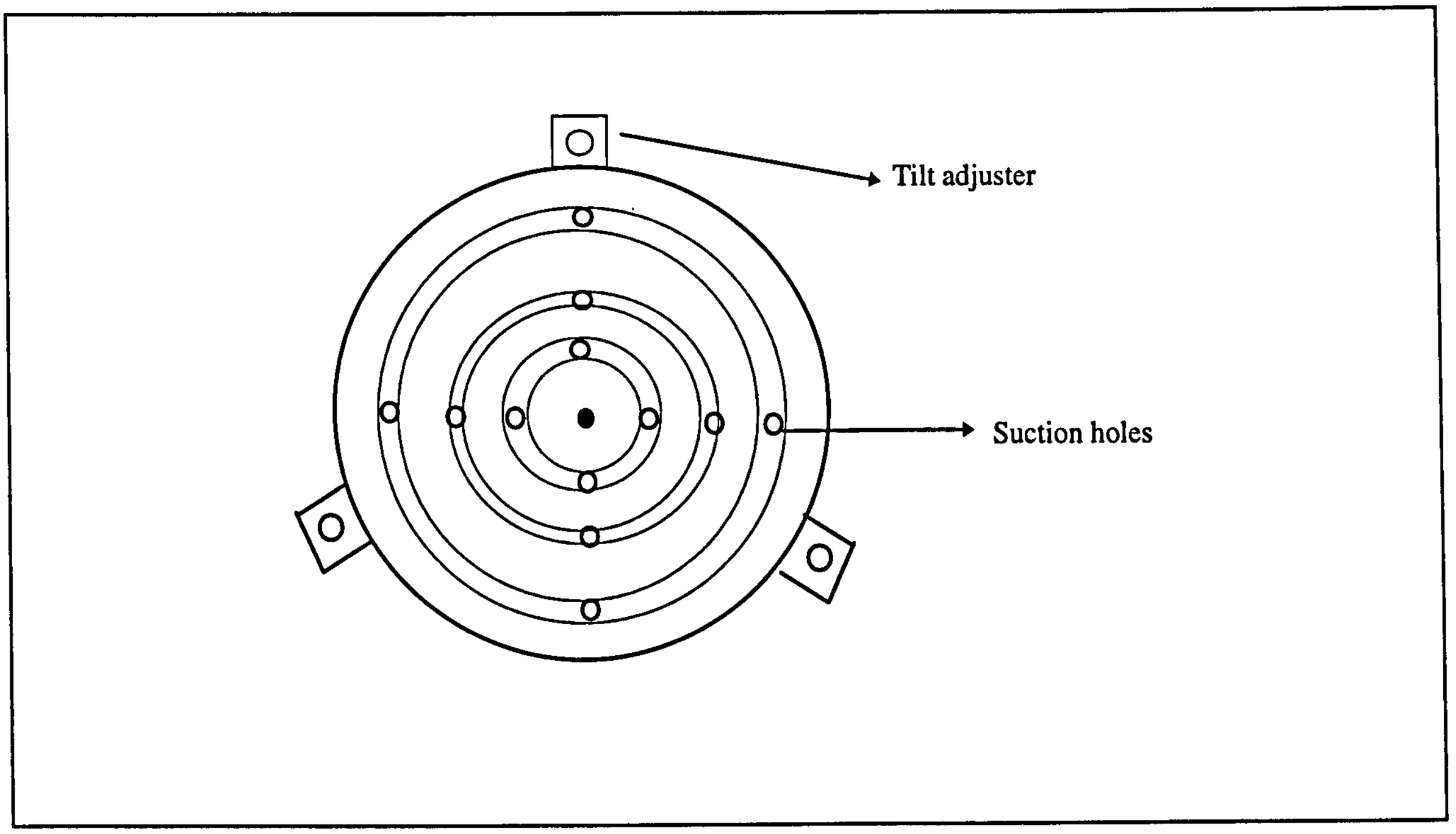


Figure 2.3 : The Vacuum chuck arrangement used on the machine

2.3 DESCRIPTION OF SPINDLE

The spindle used in this work incorporates a PI aerostatic bearing, which uses feed slots and grooves for regulating the supply pressure and bearing clearance pressure. The bearing is a combined cylindrical journal and circular thrust design (Arneson, 1969). Figure 2.4 shows a schematic of the bearing.

The bearing has the following key components:

- Lower and upper Stator
- Rotor
- Journal and thrust bearing surfaces
- Supply and exhaust slots (restrictors)
- Bearing clearance and Central Land
- Bearing grooves

A description of each component is given below (Arneson, 1969):

Lower and upper stator: According to Arneson, (1969), the upper stator forms the journal bearing surfaces within the bearing bore, it also incorporates one exhaust and two supply slots both machined on to the upper bearing stator. On the other hand the lower bearing stator is supported on the outer structure of the machine as shown in figure 2.4.

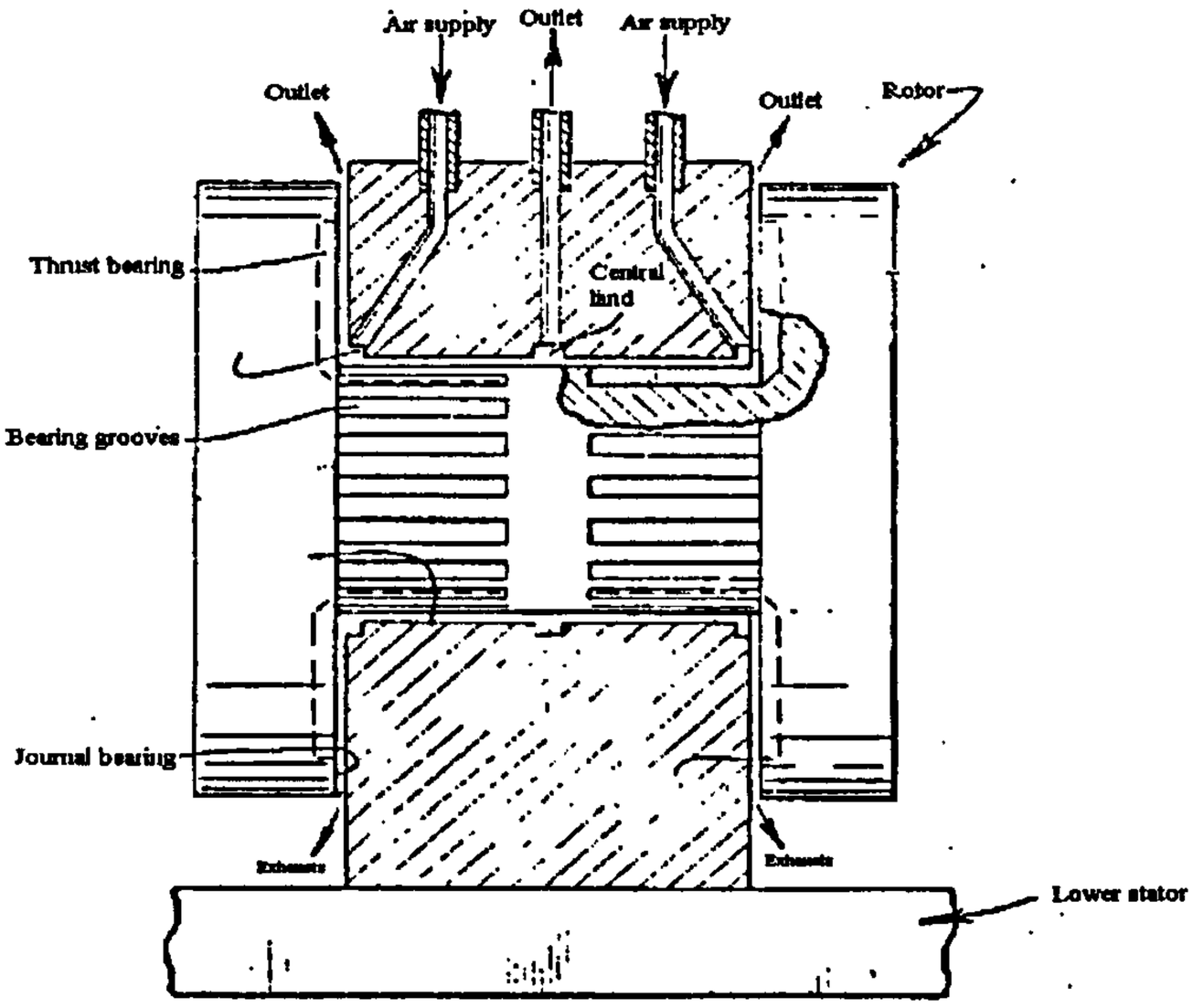


Figure 2.4: Sectioned view of the PI bearing (after Arneson, 1969)

Rotor: The rotor consists of a shaft and two thrust plates. The thrust plates have diameters (D) designed to be greater than the length (L) of the bearing shaft, this inherently minimises the angle of tilt of the rotor so that it is less than the nominal tilt angle (ie determined by the length of the shaft and the bearing clearance).

Exhausts: The bearing exhaust is an annular groove which has an outlet formed on the upper stator (see figure 2.4). The figure also shows secondary exhausts which are formed on both sides of the thrust surfaces of lower end of the stator referred to as the *lower stator.*

Bearing clearance and central land: The shaft forms a radial clearance with the journal surfaces of the stator, such that a certain level of flow restriction is maintained. The main contributor to this flow are the *viscous* forces of the fluid; *body* forces of the fluid can be considered negligible. The criterion normally for designing these bearings is that the clearance is at least 500 times less than the bearing surface length.

Centrally located on the shaft is a passage formed by a central *land* (ie a raised step geometry) normally designed to be a least 20 times the clearance it forms with the journal surface.

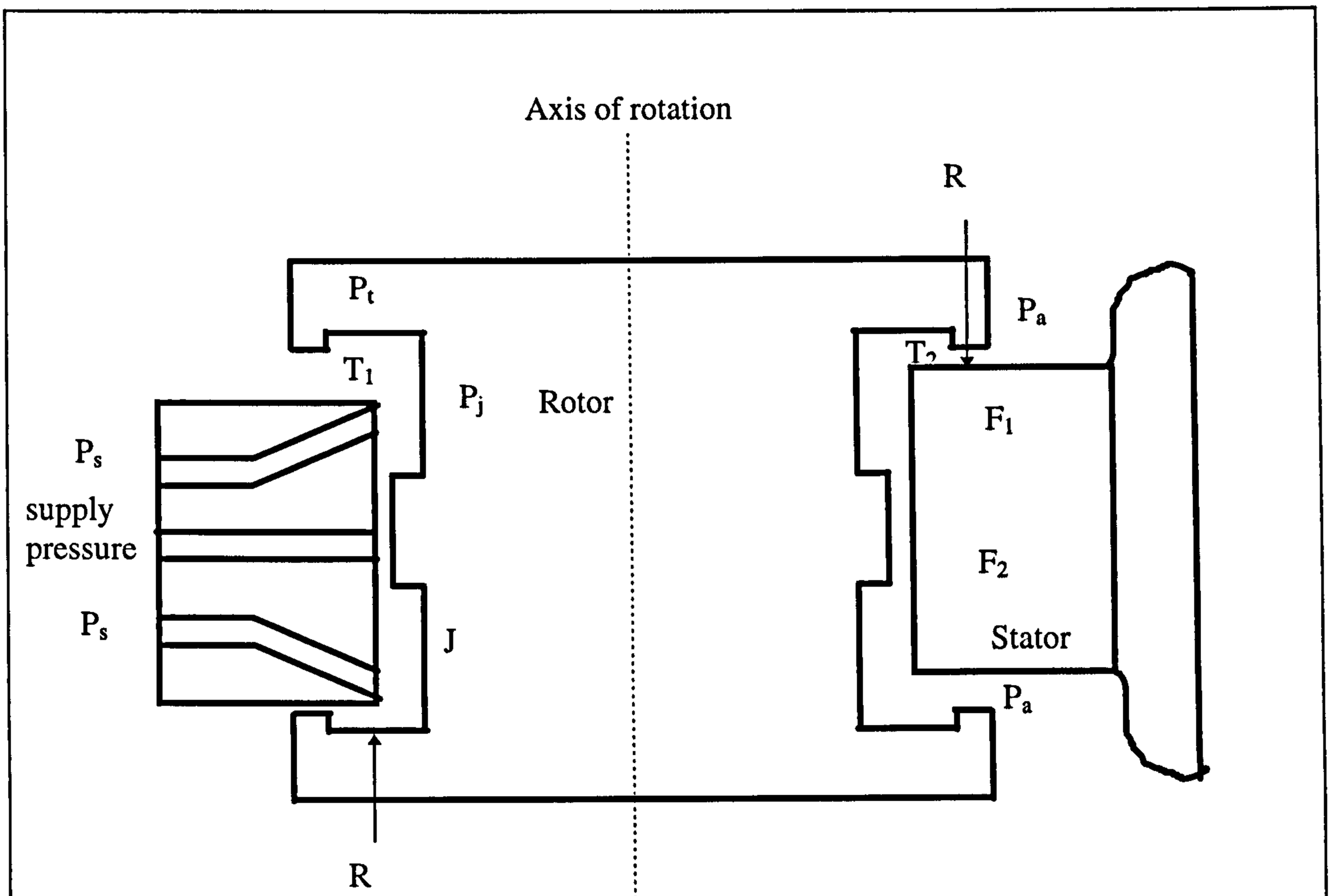


Figure 2.5: Operation of the PI bearing (after Arneson, 1969)

Bearing grooves: The bearing shaft incorporates equally spaced grooves located at either side of the central land (as shown in figure 2.4). According to Slocum (1992, pp580 - 625), the main function of bearing grooves is to control pressure distribution while maintaining a high degree of stability because squeeze film effects are maximised by long flow paths and the absence of upstream restrictors.

There has been some research carried out on the dynamic effects of different groove geometries and their influence on bearing load capacity and stability. Typical examples of this include work of Malanoski & Pan (1965), in which the static and dynamic characteristics of spiral-grooved thrust bearings were investigated. This work recognised the improvement in bearing stiffness and load capacity due to its *inflow* spiral-grooved design. Other work in this field include, Kumada et al (1996) and Kato & Obara (1996), which investigated the improved hydrodynamic lubrication and dynamic stability due to the employment of micro-grooves on the bearing surface.

A much more elaborate investigation of the physical role played by grooves in self acting bearings was made by Malanoski & Pan (1971), in which it was shown that the

pneumatic or hydraulic resistance along the grooves was smaller than that across the grooves, and that a relative sliding motion between the surfaces in a direction inclined to the grooves, tends to induce flow along the grooves. The was because the grooves behave as a *viscous pump*, and the capillary blockage of the induced flow causes pressure in the groove to rise above ambient and thereby enables the bearing to carry load. The air flow in the bearing grooves has a component normal to its axial direction, and a pressure distribution in the direction normal spindle axis of rotation.

2.4 OPERATION OF PI BEARING

The journal and thrust surfaces (J) and (T) of the bearing are supplied with line pressure P_s via two inlet restrictors located within the upper stator, as shown schematically in figure 2.5. As a result of the action of the restrictors there is a pressure P_j in the clearance space between the journal bearing and a pressure P_t in the clearance space between the thrust bearing, before the gas pressure then finally drops to ambient pressure P_a at the exit of the bearing.

According to Arneson (1969), in order to restore the spindle axis of rotation to coincide with the axis of the stator, the bearing would develop a self-restoring force in the form of a couple R-R (as shown in figure 2.5). This couple is generated by the increased pressures in the regions of decreased clearance of the thrust bearings surface labelled T_1, and a decrease in pressure in regions of increased clearance of the thrust bearing surface labelled T_2. In addition to the forces generated between the thrust surfaces, their is also an extra restoring force normal to the couple R-R generated within the clearance space between the lower stator and the journal surface. This force is labelled F_1 for regions of decreased journal clearance and F_2 for regions of increased journal clearance.

2.5 CHARACTERISTICS OF PI BEARING

2.5.1 INTRODUCTION

In this section, a summary of the bearing performance is given. There will also be a brief description of some the design considerations of the bearing, particularly related to the air flow regimes between bearing (journal and thrust) surfaces, and within its supply restrictor.

2.5.2 BEARING PERFORMANCE DATA

Tables 2.1a - 2.1c show the general performance of the PI 4R air bearing used in this work. This data relates to the bearing performance at 1034 KPa (150 psi) supply pressure, however according to the spindle manufacture (see appendix A), the bearing is capable of running successfully from less than 344 KPa (50 psi) to 1378 KPa (200 psi) without showing any signs of instability.

Tables 2.1a and 2.1b show the load capacity and stiffness of the spindle in its axial, radial and tilt degrees of freedom. The load capacity and stiffness are approximately linear with air pressure. The *ultimate* load capacity shown in table 2.1b is measured (by the bearing manufacturer) when metal-to-metal contact is made across the bearing, while the *working* load is usually given as half the ultimate load capacity.

LOAD CAPACITY	*ULTIMATE*	*WORKING*
RADIAL	445 N	222.5 N
AXIAL	1780 N	890 N
TILT	48 Nm	22.5 Nm

Table 2.1: Spindle load carrying capacity data according to bearing manufacturer

STIFFNESS	
RADIAL	117 $N\mu m^{-1}$
AXIAL	350 $N\mu m^{-1}$
TILT	0.42 $Nm\ \mu rad^{-1}$

Table 2.2: Spindle load stiffness data according to bearing manufacturer

GENERAL FEATURES	
TOTAL WEIGHT	8.2 Kg
ROTOR WEIGHT	3.5 Kg
ROTOR INERTIA	5160 gcm^2
MAXIMUM SPEED	10, 000 RPM
AIR FLOW RATE	$< 0.057\ m^3min^{-1}$

Table 2.3: General bearing data provided by bearing manufacturer

According the bearing manufacturer (see attached data sheet for more details), some of the general features of the bearing include:

- bearing has the capability of providing resistance to tilt (ie tilt stiffness) or other loads, as a result of its combined restoring forces along the journal and thrust surfaces.

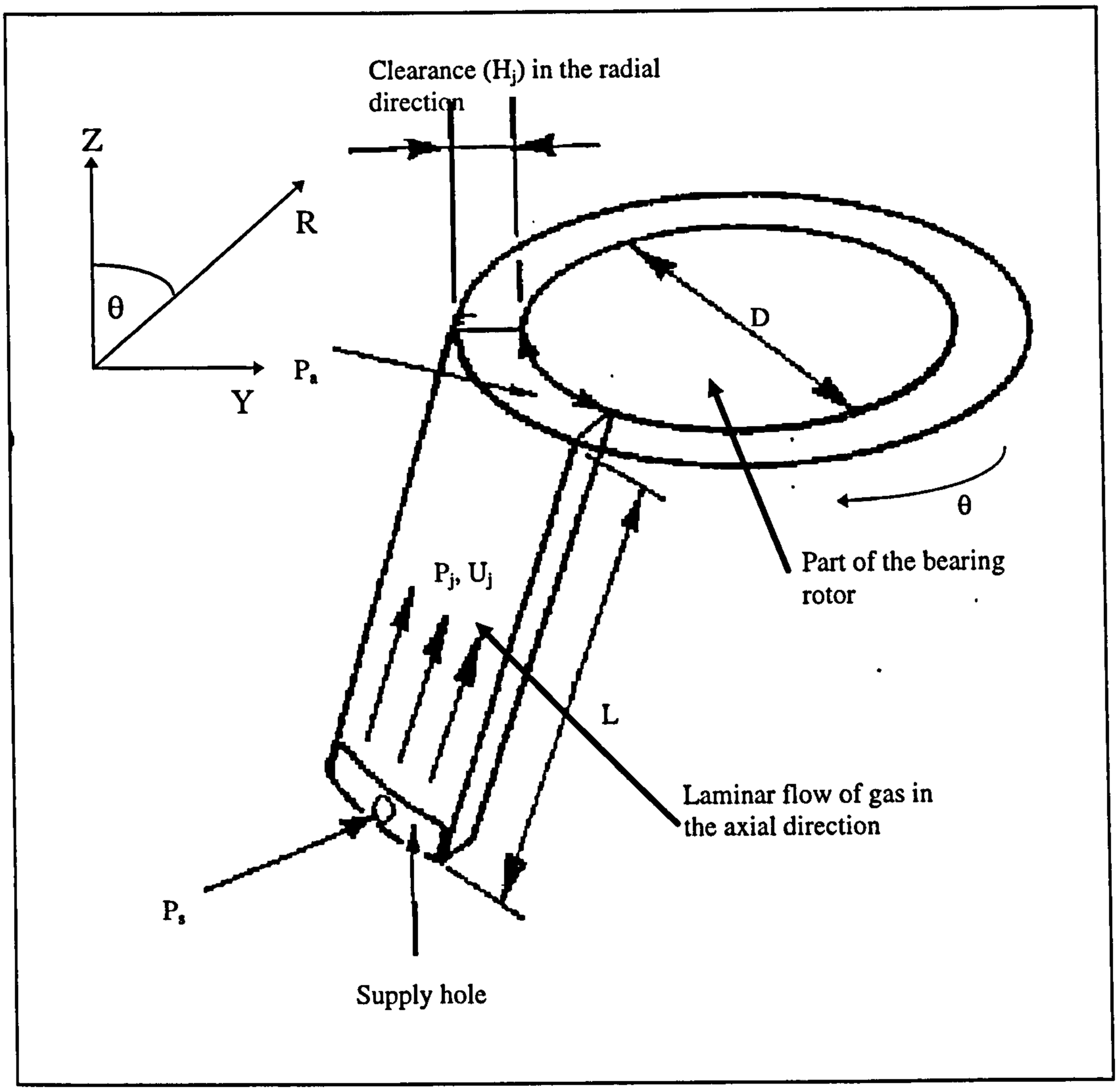


Figure 2.6: Model used to describe flow in the PI bearing journal surfaces (after Powell, 1970)

- reduced angular errors due to the geometry of the thrust plates and shaft (ie the ratio of the diameter of the thrust plate to the length of the shaft is greater than one).

- reduced bearing static and dynamic instabilities as a result of the suppressed squeeze film effects, due to bearing axial grooves.

- increased stiffness (static and dynamic) and load capacity of the bearing due to full pressure supply on both the journal and thrust surfaces.

2.5.3 EQUATIONS OF FLOW BETWEEN BEARING SURFACES

In this section some basic equations which describe the flow regime in the PI bearing are presented. As air flows between bearing journal and thrust surfaces, pressures (P_j)

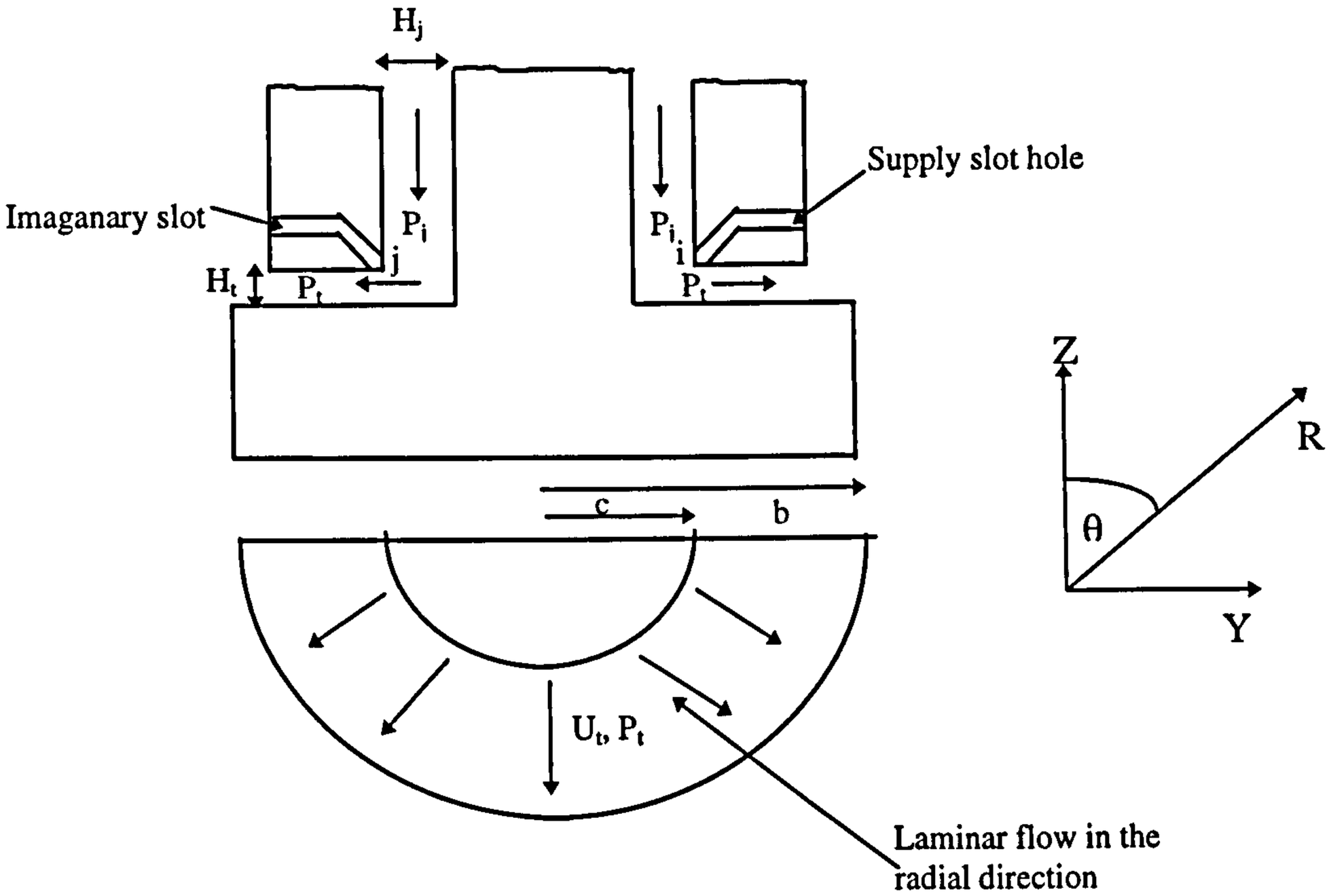


Figure 2.7: Sectioned diagram of PI thrust plates (after Powell, 1970)

and (P_t) in the spindle axial (Z), radial (R) and circumferential (θ) positions are generated (ie see figures 2.6 and 2.7) .

An analytical relationship between the pressure gradients, and journal (H_j) and thrust (H_t) clearances is given by Reynolds' equations of bearing films (DiPrama, 1971), these are:

$$\frac{\delta}{\delta\theta}\left(H_j^3 P \frac{\delta P_j}{\delta\theta}\right)+\frac{\delta}{\delta Z}\left(H_j^3 P \frac{\delta P_j}{\delta Z}\right)=\Lambda\frac{\delta\left(P_j H_j\right)}{d\theta} \qquad (2.1)$$

$$\frac{1}{R^2}\frac{\delta}{\delta\theta}\left(H_t^3 P \frac{\delta P_t}{\delta\theta}\right)+\frac{1}{R}\frac{\delta}{\delta R}\left(H_t^3 P R \frac{\delta P_t}{\delta Z}\right)=\Lambda\frac{\delta\left(P_t H_t\right)}{d\theta} \qquad (2.2)$$

respectively.

The formulation of equations 2.1 and 2.2 is based on three classical fluid mechanics equations, namely; the *momentum*, *continuity* and *Navier-Stokes* equations. The compressibility number Λ (given in equations 2.1 and 2.2) is a unique parameter

which relates to the load carrying capacity of *aerostatic* or *aerodynamic* bearings, and the degree of compressibility required to sustain it. Experiments have shown that as Λ increases H_j (or H_t) decreases (Williams 1994), this effect inherently increases the load carrying capacity of the bearing. Λ also relates to the rotor-dynamics of the spindle in terms of oscillatory sources of error motions.

The gas flow between the bearing journal or thrust surfaces can be idealised as laminar flow between two parallel plates, as long as the following assumptions are made:

- The flow through the bearing is considered to be purely viscous with no slip boundaries, and with the inertial forces due to acceleration neglected (Slocum 1992, pp580 - 625).

- Pressure is constant over any section normal to the flow direction ie across the clearance of the journal or thrust surface (Powell 1970).

- Isothermal conditions are assumed throughout the bearing gas film. (Slocum 1992, pp580 - 625).

With the aid of one-dimensional Navier-Strokes equations, the above assumptions can be used to describe the relationship between the bearing air velocities (U_j) and (U_t) (ie acting in the radial and axial directions respectively), and their respective pressure (P_j) and (P_t) gradients:

$$\frac{d^2 U_j}{dY^2} = \frac{1}{\eta}\frac{dP_j}{dX} \tag{2.3}$$

$$\frac{d^2 U_t}{dZ^2} = \frac{1}{\eta}\frac{dP_t}{dR} \tag{2.4}$$

Figure 2.6 shows a schematic of air flow within the journal bearing. Powell (1970) showed that by integrating equation 2.3 twice and substituting the relevant boundary conditions, the theoretical mass flow rate between the journal surfaces is:

$$\dot{M}_j = \frac{\left(P_s^2 - P_a^2\right)\pi D\, H_j{}^3}{12\eta RTL} \tag{2.5}$$

Where,

P_s: bearing supply pressure
P_a: atmospheric pressure
D: diameter of bearing
R: universal gas constant
T: absolute temperature
L: bearing length
η: viscosity of bearing fluid

Application of equation 2.5 to the PI bearing assumes that it has smooth journal surfaces, modifications of these equation is required in order to accommodate the grooved nature of the bearing surface.

Figure 2.7 shows a sectioned diagram of bearing thrust plates and supply feed slots. One of the depicted feed slots shown in top diagram of the figure is actually imaginary, but it was drawn to illustrate the symmetry in operation of the circular thrust plate, (since the pressure at i and j are designed to be approximately the same). The theoretical mass flow rate between the thrust surfaces ie evaluated from equation 2.4 (Powell, 1970) is:

$$\dot{M}_t = \frac{\left(P_s^2 - P_a^2\right)\pi H_t^3}{12\eta RT \log\left(\frac{c}{b}\right)} \qquad (2.6)$$

Where,

c: minor radius of thrust plate (as shown in figure 2.7)
b: major radius of thrust plate

2.5.4 EQUATIONS OF FLOW FOR BEARING RESITRICTOR (FEED SLOT)

Restrictors are machined in the wall of the bearing for supplying gas at line pressure to the journal and thrust surfaces. Figure 2.8 shows the geometry of the restrictor used in the PI bearing, which is an annular orifice. Some sort of flow resistance is required in the bearing operation in order to maintain differential pressure, and thus giving the bearing nearly zero compliance.

In this section, equations which describe the flow of air in the bearing restrictors are given. In figure 2.8 gas accelerates through the *flow area* (πdh) from the bearing supply pressure (P_s) to a reduced pressure (P_r) (ie known as the orifice throat pressure) as it approaches the bearing surface. Powell (1970) studied the flow through an orifice restrictor of uniform geometry, and gave the relationship between P_s and P_r as:

$$\frac{P_r}{P_s} = \left[1 - \frac{\alpha - 1}{2}\left(\frac{v}{a_o}\right)^2\right]^{\frac{\alpha}{\alpha-1}} \qquad (2.7)$$

where,

v: velocity of flow within the slot
a_o: speed of sound at the supply conditions
α: ratio heat capacities at constant pressure and volume

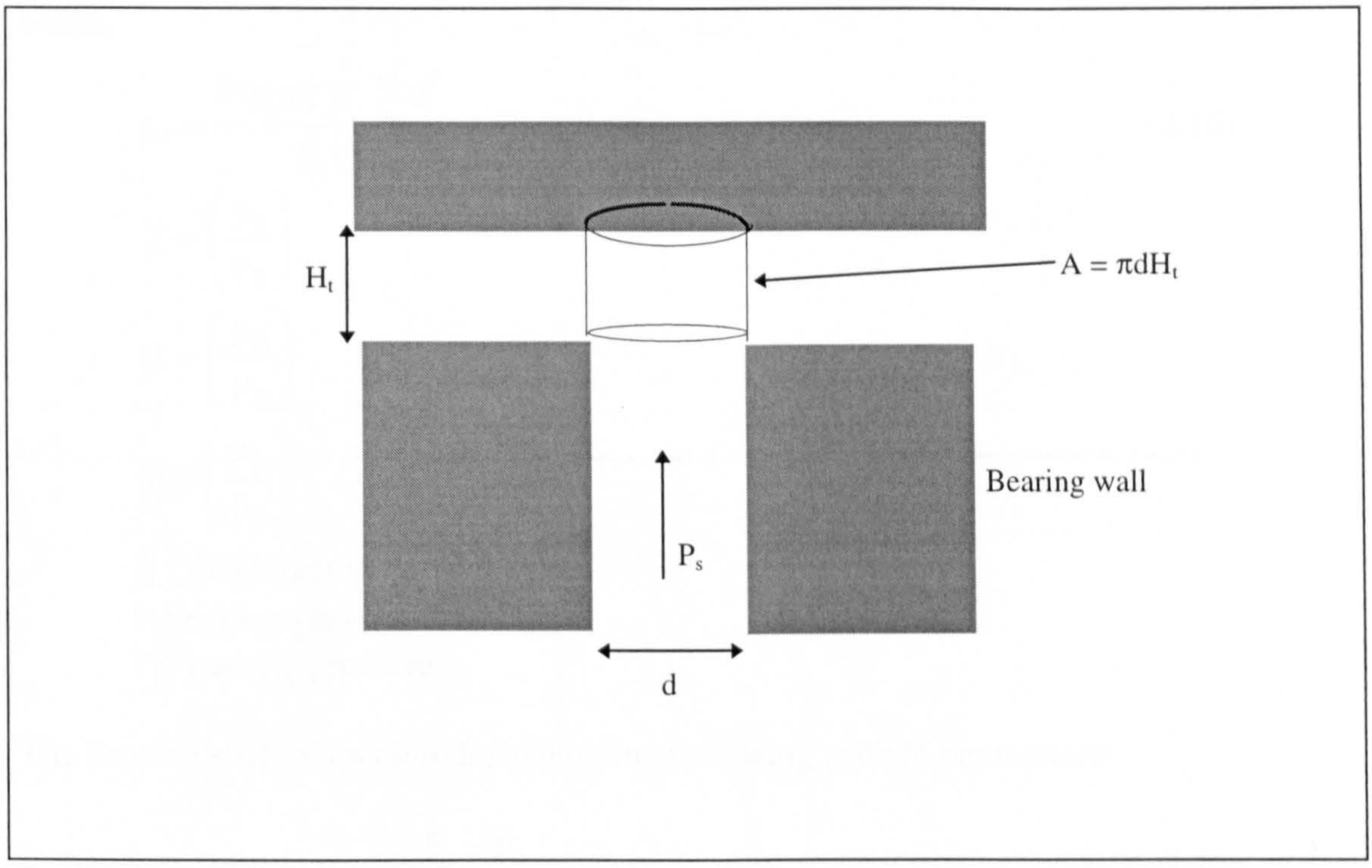


Figure 2.8: Geometry of the PI bearing feed slot (after, Powell 1970)

Assuming isentropic expansion of the gas within the bearing, the mass flow within an annular orifice according to Slocum (1992, pp580 - 625) is:

$$\dot{M}_{orifice} = \pi d H_t C_d \left[\frac{2\alpha}{(\alpha-1)RT} \left\{ \left(\frac{P_r}{P_s} \right)^{\frac{2}{\alpha}} - \left(\frac{P_r}{P_s} \right)^{\alpha+\frac{2}{\alpha}} \right\} \right]^{\frac{1}{2}} \qquad (2.8)$$

where,

d: diameter of annular orifice
H_t: clearance of bearing
C_d: flow discharge coefficient
R: universal gas constant
T: absolute temperature

Slocum (1992, pp580 - 625) also equated the mass flow for (N) number of orifices by modifying equation 2.8, giving:

$$\Lambda_S \xi C_d \left[\frac{2\alpha}{(\alpha-1)} \left\{ Z^{\frac{2}{\alpha}} - Z^{\alpha+\frac{2}{\alpha}} \right\} \right]^{\frac{1}{2}} = U^2 - V^2 \qquad (2.9)$$

where,

$$\Lambda_s = \frac{6\eta (RT)^{\frac{1}{2}}\ N d^2}{P_s h^3}$$: orifice feeding parameter (2.10)

$$Z = \left(\frac{P_t}{P_S}\right)$$

$$U = \left(\frac{P_o}{P_S}\right)$$

$$V = \left(\frac{P_b}{P_S}\right)$$

ξ : shape factor

P_o: orifice pressure

P_b: bearing pressure

The bearing stiffness would depend on the following critical parameters:

- *pressure ratio*; $K_{go} = \frac{P_o - P_b}{P_s - P_b}$ (2.11)

- *orifice feeding parameter* (defined in equation 2.10)

Flow equations derived in this chapter were used in section 3.5 to estimate the error motions of the spindle bearing.

CHAPTER 3: GENERAL CHARACTERISTICS OF SPINDLE ERROR MOTIONS

3.1 INTRODUCTION

Ideally, a spindle should rotate about a *true* single axis of rotation; ie it should only have one degree of freedom (DOF). But generally rigid bodies have six DOFs as shown in figure 3.1, including three translational motions and three angular motions. Hence one would expect an imperfect spindle to revolve about an axis that is different from the true axis of rotation due to its radial $\mathbf{d_x}$ and $\mathbf{d_y}$, axial $\mathbf{d_z}$ and tilt $\mathbf{E_x}$ and $\mathbf{E_y}$ error motions. These errors-motions can be described as the *parasitic motions*. as Jones (1956) used in respect of micro-displacements

In this chapter, a mathematical model is developed to describe spindle error motions with the aid of homogeneous transformation matrixes. A theoretical analysis has been used to estimate the error motions of the spindle using the bearing stiffness and damping constants. A description of the sources of spindle error motions is also presented.

3.2 DEFINITION OF SPINDLE ERROR MOTIONS

The following definitions are based on the spindle analysis work carried-out by Bryan et al (1967), ANSI/ASME B89.3.4M (1985) and Slocum (1992, pp62 - 70), refer to figure 3.1 for more details:

- **radial error motion (d_x, d_y)**: error motion in a direction normal to the Z reference axis and at a specified axial location.

- **axial error motion (E_z)**: error motion colinear with the Z reference axis.

- **tilt error motion (E_x, E_y)**: error motion in an angular direction relative to the axial Z reference axis. If the tilt motion along the Y axis is in the *sensitive direction*, a corresponding error motion will occur about the X axis (ie which is assumed to be a gauging or machining axis).

- **sensitive directions:** the sensitive direction is perpendicular to the ideally generated workpiece surface through the instantaneous point of machining or gauging. The *rotating sensitive direction* is where a workpiece is fixed and point of machining or gauging rotates with the spindle. A *fixed sensitive direction* on the other hand is where a workpiece is rotated by the spindle and the point of matching or gauging is fixed.

- **run-out error**: is the total error motion (displacement) measured by a device that senses a rotating artefact surface (ie mounted on the test spindle), or is rotated with respect to a fixed surface of the artefact.

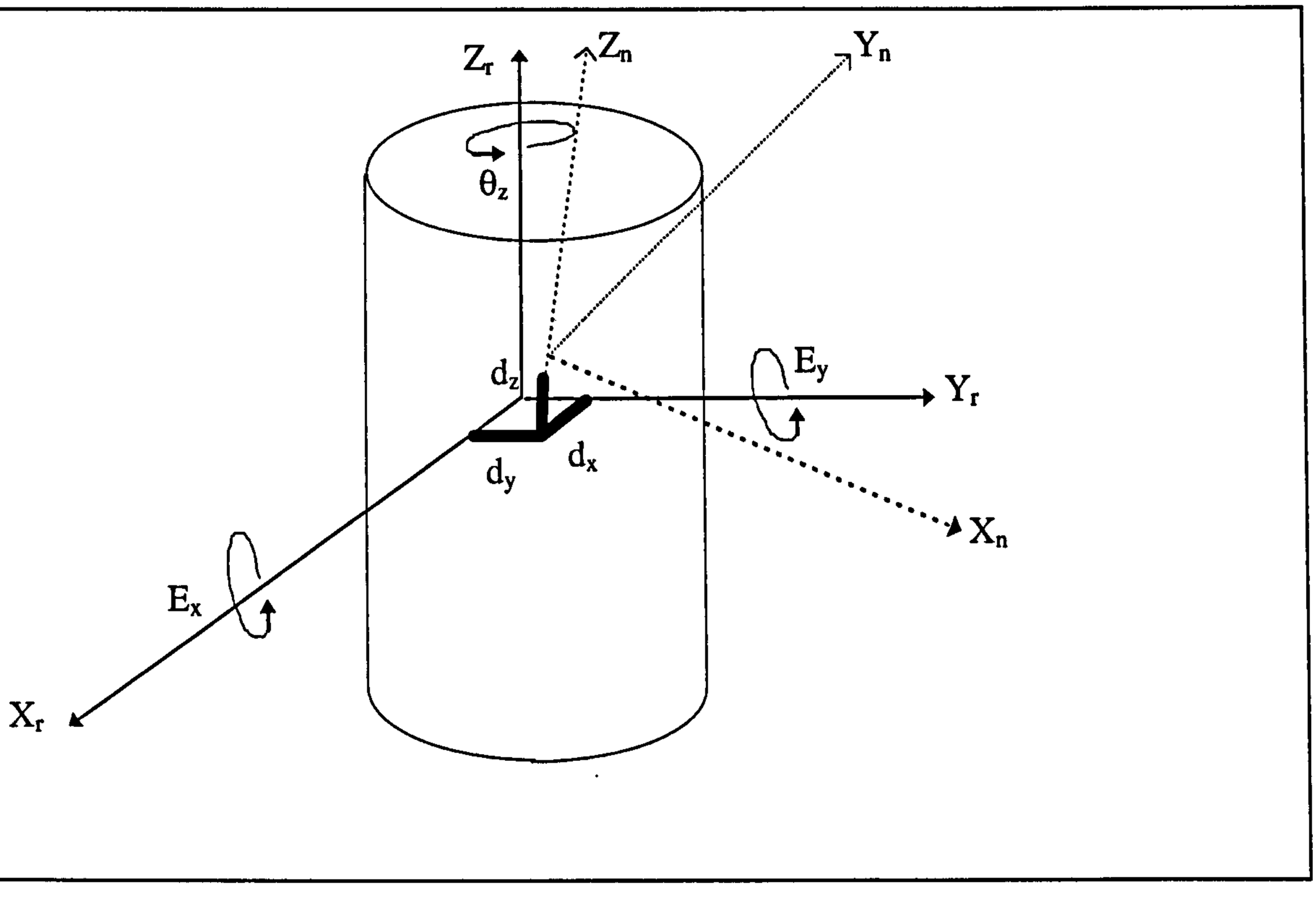


Figure 3.1: Error motions associated with a spindle

- **average error motion:** is the mean value of a series of *total error motion* values

- **asynchronous error:** is the *total error motion* measured and is equal to the run-out error of the spindle when the error displacements occur at frequencies other than the fundamental or harmonic frequencies of rotation of the spindle.

- **synchronous error:** is the difference between the *total error motion* and the *average error motion* values, and is equal to the run-out of the spindle when the displacements occur at the rotational frequency of the spindle.

- **error motion polar plot:** sometimes referred to as a *rosette* diagram, is a polar co-ordinate representation of error motion magnitude at specified angular positions (in degrees)

3.3 HTM MODEL FOR SPINDLE ERROR MOTIONS

In order to describe the error components of a spindle in three dimensional space with respect to a reference co-ordinate system, a 4 x 4 matrix is required. A similar representation is used to study the dynamics of mechanical systems, (Stadler, 1995).

This homogeneous transformation matrix may be used to represent the co-ordinate transformation of say a point on the spindle co-ordinate frame X_n, Y_n, and Z_n to the reference co-ordinate system X_r, Y_r, and Z_r and according to Slocum (1992, pp62 - 70) the transformation matrix can be represented as follows:

$$T_e = \begin{bmatrix} O_{ix} & O_{iy} & O_{iz} & P_x \\ O_{jx} & O_{jy} & O_{jz} & P_y \\ O_{kx} & O_{ky} & O_{kz} & P_z \\ 0 & 0 & 0 & P_s \end{bmatrix} \qquad (3.1)$$

The first three columns of the above matrix simply define the orientation of the spindle X_n, Y_n, and Z_n axes with respect to the reference frame, in the form of direction cosine unit vectors i, j and k. Whereas the last column represents the position of the origin of the spindle co-ordinate system with respect to the reference frame. P_s is a scale factor, which is normally set to unity. Hence the transformation of a point on the spindle with respect to the reference frame is:

$$\begin{bmatrix} X_r \\ Y_r \\ Z_r \\ 1 \end{bmatrix} = T_e \begin{bmatrix} X_n \\ Y_n \\ Z_n \\ 1 \end{bmatrix} \qquad (3.2)$$

By considering all five error motions of the spindle as functions of the angle of rotation using the HTM model, and by setting n = 1; the respective error motions can be expressed as follows:

Radial error d_x will be due to the translation of the X_1, Y_1, and Z_1 co-ordinate system by an amount d_x along the X axis. So that the HTM (homogeneous transformation matrix) of the X_1, Y_1, and Z_1 is given as:

$$T_{d_x} = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (3.3)$$

Radial error d_y is due to the translation of the X_1, Y_1, and Z_1 co-ordinate system by the amount d_y along the Y axis, this transformation can be represented as:

$$T_{d_y} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (3.4)$$

Axial error d_z is due to the translation of X_1, Y_1, and Z_1 co-ordinate system by the amount d_z along the Z axis, this transformation can be represented as :

$$T_{d_z} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (3.5)$$

Tilt error E_x is due to the rotation of the X_1,Y_1, and Z_1 co-ordinate system by an amount E_x of the tilt about the X axis (as shown in figure 3.2), this transformation can be represented as:

$$T_{E_x} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos E_x & -\sin E_x & 0 \\ 0 & \sin E_x & \cos E_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (3.6)$$

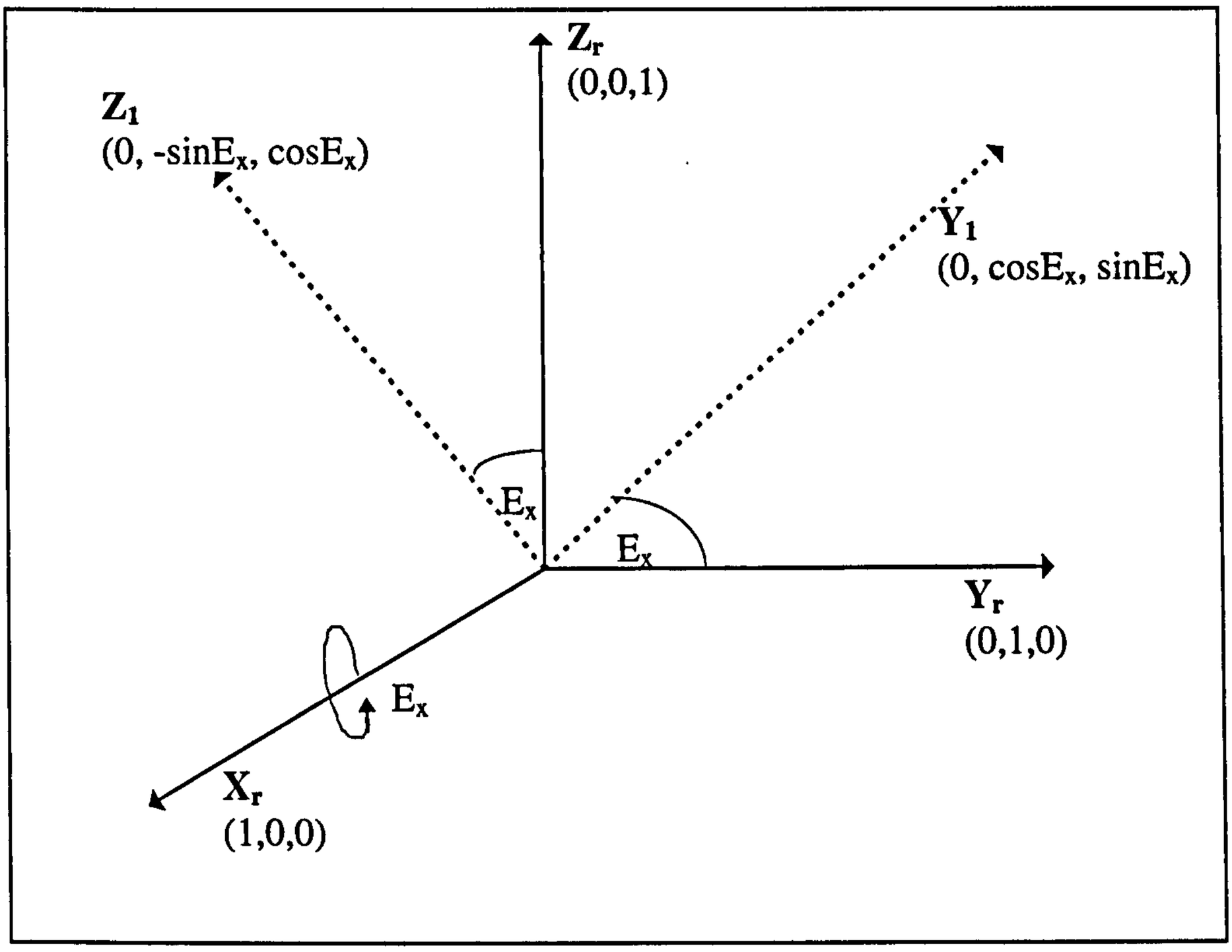


Figure 3.2: HTM representation of tilt error E_x

Pitch error E_y is also due to the rotation of the X_1, Y_1, Z_1 co-ordinate system by an amount E_y of tilt about the Y axis, (as shown in figure 3.3), this transformation can be represented as:

$$T_{E_y} = \begin{bmatrix} \cos E_y & 0 & \sin E_y & 0 \\ 0 & 1 & 0 & 0 \\ -\sin E_y & 0 & \cos E_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (3.7)$$

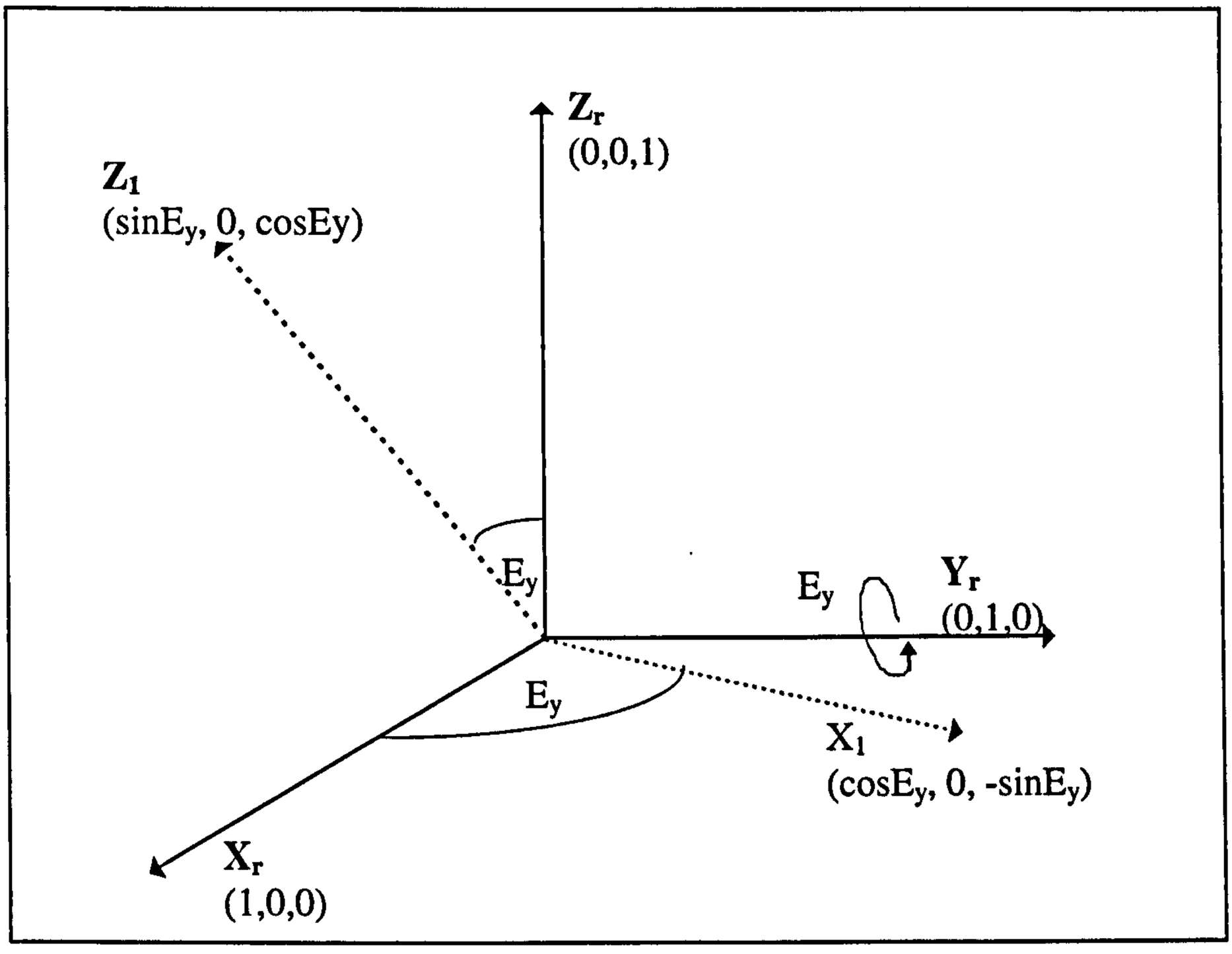


Figure 3.3 : HTM representation of the tilt error E_y

Yaw component of the spindle O_z is due to the rotation of the X_1, Y_1 and Z_1 co-ordinate system by an amount O_z about the Z axis (as shown in figure 3.4), this transformation can be represented as:

$$T_{\theta_z} = \begin{bmatrix} \cos\theta_z & \sin\theta_z & 0 & 0 \\ -\sin\theta_z & \cos\theta_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (3.8)$$

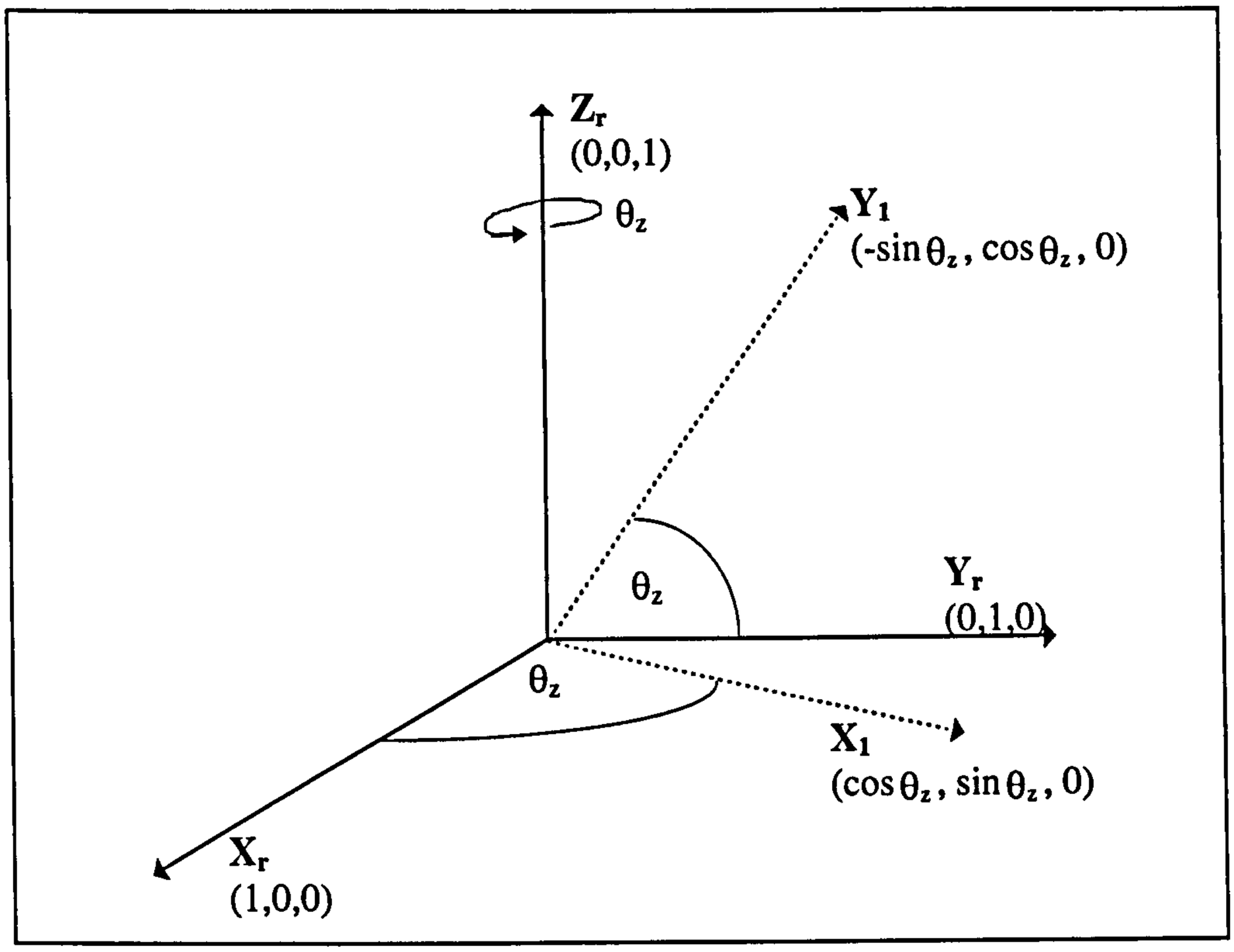


Figure 3.4 : HTM representation of the spindle rotation O_z

The total effect of the spindle error motions is evaluated by sequentially multiplying the respective error HTMs and the yaw component of the spindle as follows:

The total error, T_e

$$= [T_{d_x}] \cdot [T_{d_y}] \cdot [T_{d_z}] \cdot [T_{E_x}] \cdot [T_{E_y}] \cdot [T_{\theta_z}] \qquad (3.9)$$

$$= \begin{bmatrix} \cos E_y \cos\theta_z & -\cos E_y \sin\theta_z & \sin E_y & d_x \\ \sin E_x \sin E_y \cos\theta_z + \cos E_x \sin\theta_z & \cos E_x \cos\theta_z - \sin E_x \sin E_y \sin\theta_z & -\sin E_x \cos E_y & d_y \\ -\sin E_x \sin\theta_z + \cos E_x \sin E_y \cos\theta_z & \sin E_x \cos\theta_z + \cos E_x \sin E_y \sin\theta_z & \cos E_x \cos E_y & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (3.10)$$

According to Slocum (1992, pp62 - 70) when sub-micrometre spindle accuracy levels are required, the second order error components can be neglected. This is because angles E_x and E_y are so small, hence $\cos E_x \sim \cos E_y \sim 1$, $\sin E_x \sim E_x$ and $\sin E_y \sim E_y$.

So T_e can be simplified to:

$$\begin{bmatrix} \cos\theta_z & -\sin\theta_z & E_y & d_x \\ \sin\theta_z & \cos\theta_z & -E_x & d_y \\ E_x\sin\theta_z - E_y\cos\theta_z & E_x\cos\theta_z + E_y\sin\theta_z & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (3.11)$$

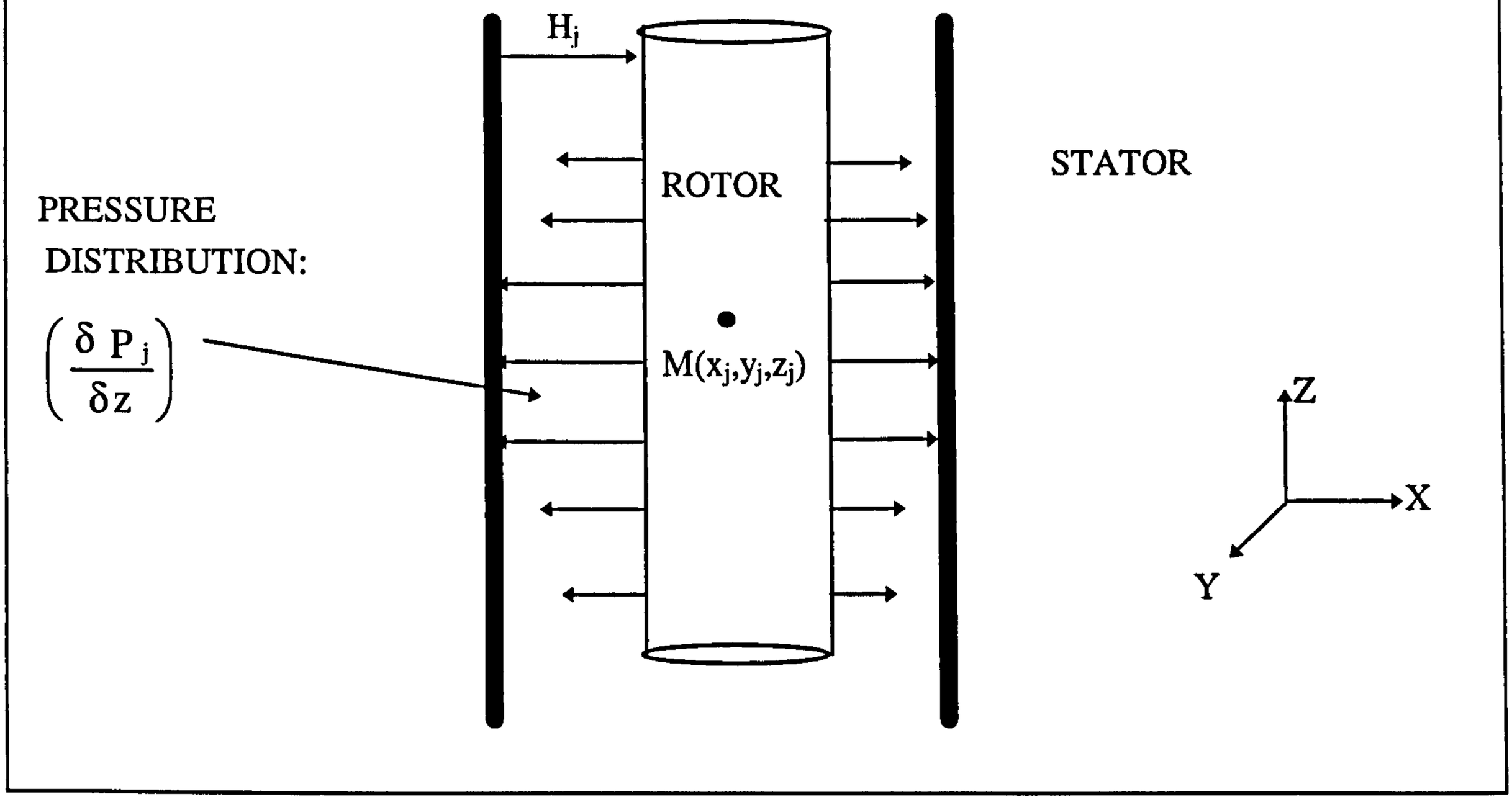
3.4 EFFECT OF ERROR MOTION ON BEARING CLEARANCE AND PRESSURE GRADIENT

As described in section 2.5.2, the spindle used in the project is essentially a combined journal and thrust bearing arrangement. Error motions of a spindle will transform its reference co-ordinate system to a new co-ordinate system, this will inherently perturb its clearance, pressure gradient and dynamic coefficients (bearing stiffness and damping constants). The homogeneous transformation matrix T_e of the combined error motions used to make this transformation is given in equation 3.11.

A simple model can be used to evaluate the effects of spindle error motions. Such a model was developed by Iordanoff et al (1995) and used for analysing the effect of misalignment and load on thrust bearings. Klit and Lund (1986), also developed a similar model for calculating the dynamic coefficients (film stiffness and damping constants) of journal aerodynamic bearings using a finite element *variational approach*. Modelling the effects of error motions in spindles requires the substitution of the perturbed bearing film clearance (h) and pressure (p) terms in Reynolds differential equation for compressible fluid (Han et al). This differential equation is then solved analytically or numerically using a number of methods which include, *PH pertubation*, *linearized PH*, finite-difference and finite-element discretisation (Gross et al, 1980). In this section equations which describe the perturbation of bearing clearance and pressure gradient are given.

Figure 3.5 shows a schematic of a journal bearing model in its reference frame, indicating the radial bearing clearance H_j and pressure distribution

$\left(\frac{\delta P_j}{\delta z}\right)$.



Figures 3.5: Model Of Journal Bearing

By using equation 3.11, the perturbed journal bearing clearance with centroid position $M_j(x,y,z)$ due to the influence of spindle error motions can be defined as:

$$[H_j] = H_j^0 + H_j^1 = H_j^0 + [T_e] \cdot [M_j] \qquad (3.12)$$

where,

$$[T_e] = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & E_y & d_x \\ \sin\theta_z & \cos\theta_z & -E_x & d_y \\ E_x\sin\theta_z - E_y\cos\theta_z & E_x\cos\theta_z + E_y\sin\theta_z & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$: Error motion matrix

$$M_j = \begin{bmatrix} x_j \\ y_j \\ z_j \\ 1 \end{bmatrix}$$: Centroid position vector of journal bearing

H_j^0 : Nominal journal bearing clearance at steady state

H_j^1 : Bearing clearance due to error motion contributions

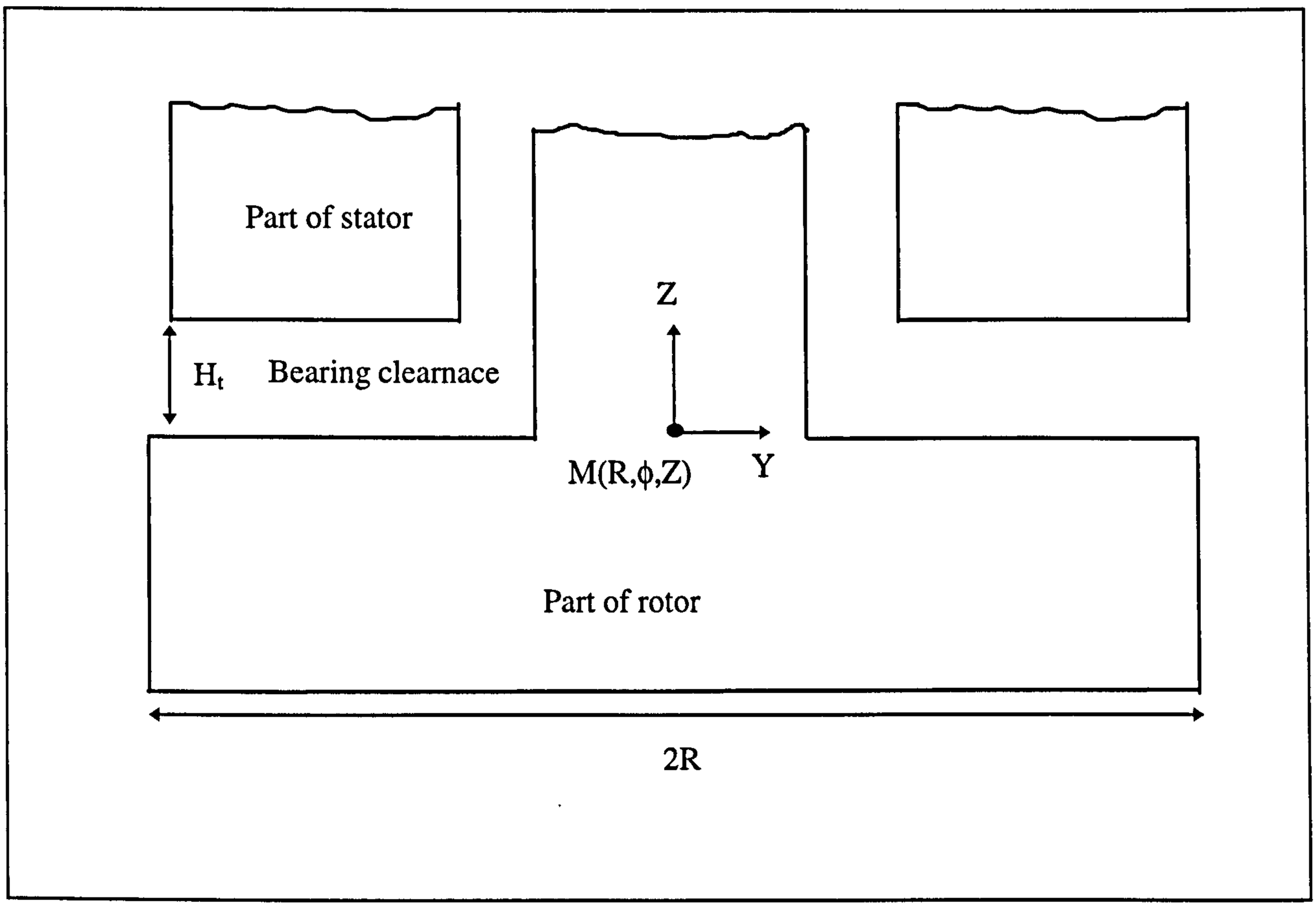


Figure 3.6a: Section view Of Thrust Bearing Model

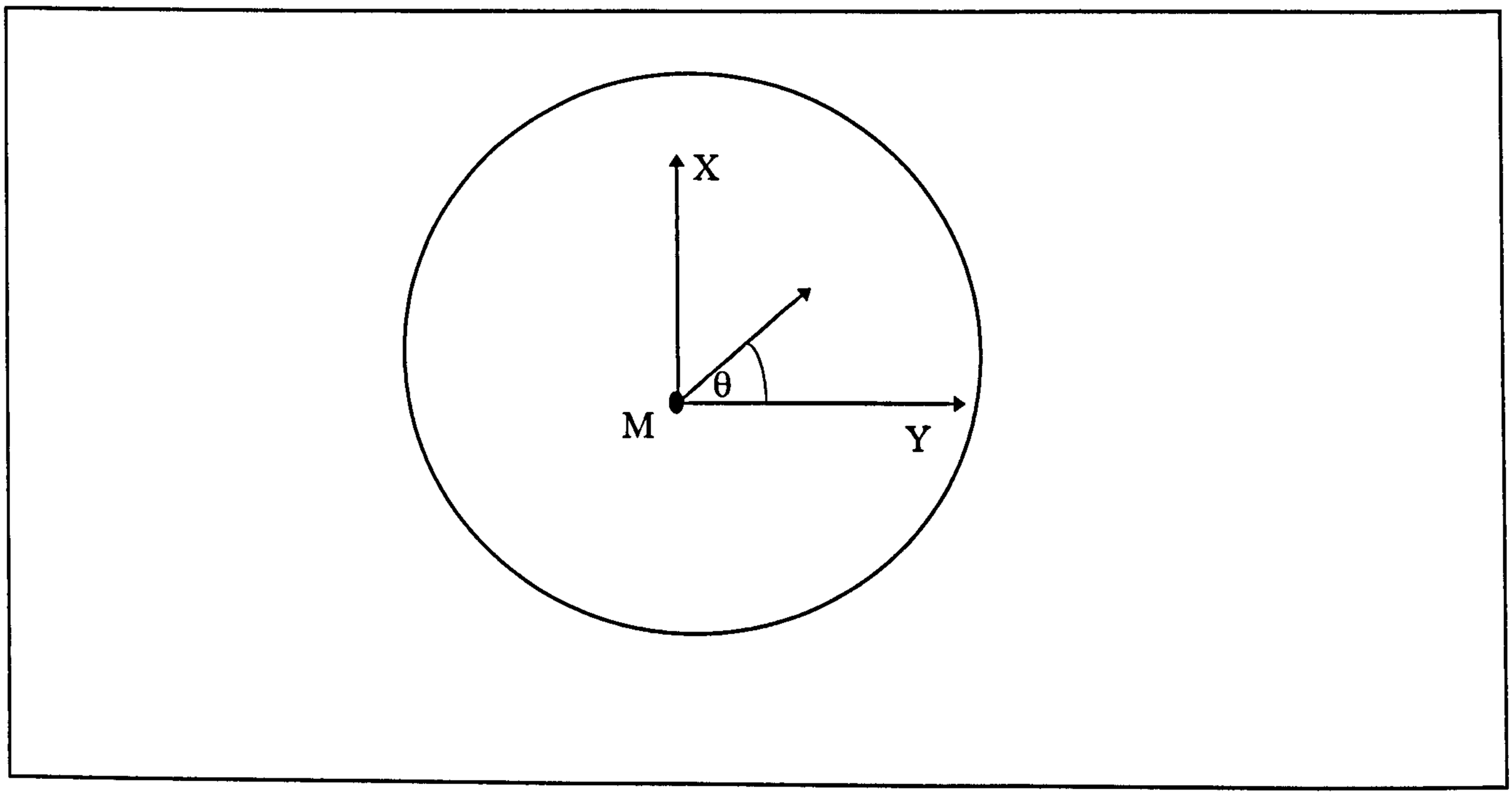


Figure 3.6b: Plan View Of Thrust Bearing Model

hence by simplifying equation 3.12,

$$H_j^1 =$$

$$\begin{bmatrix} x_j\cos\theta_z + y_j\sin\theta_z + z_j E_y + d_x \\ x_j\sin\theta_z + y_j\cos\theta_z - z_j E_x + d_y \\ x_j(E_x\sin\theta_z - E_y\cos\theta_z) + y_j(E_x\cos\theta_z + E_y\sin\theta_z) + z_j + d_z \end{bmatrix} \qquad (3.13)$$

Figure 3.6a and 3.6b depicts the model used for describing the effects of spindle error motions of a thrust bearing with centriod position $M_t(r, \phi, z)$ (ie based on a polar co-ordinate system). The perturbed thrust bearing clearance is:

$$[H_t] = H_t^0 + H_t^1 = H_t^0 + [T_e] \cdot [M_t] \qquad (3.14)$$

where,

$$M_t = \begin{bmatrix} r_t\sin\phi \\ r_t\cos\phi \\ z_t \\ 1 \end{bmatrix} : \text{Centroid position vector of thrust bearing}$$

H_t^0: Nominal thrust bearing clearance at steady state
H_t^1: Error motion contributions of journal bearing clearance

hence $H_t^1 =$

$$\begin{bmatrix} r_t\sin\phi\cos\theta_z - r_t\cos\phi\sin\theta_z + z_t E_y + d_x \\ r_t\sin\phi\sin\theta_z + r_t\cos\phi\cos\theta_z - z_t E_x + d_y \\ r_t\sin\phi(E_x\sin\theta_z - E_y\cos\theta_z) + r_t\cos\phi(E_x\cos\theta_z + E_y\sin\theta_z) + z_t + d_z \end{bmatrix} \qquad (3.15)$$

The perturbed pressure distribution of the bearing is due to the small perturbations of the thrust and journal clearances (H_t^1 and H_j^1), and their first derivatives ($\dot{H}_t^1$ and $\dot{H}_j^1$). A first order expansion of the journal and thrust pressures are Klit and Lund (1986):

$$P_j = P_j^0 + P_j^1 H_j^1 + P_j^1 \dot{H}_j^1 \qquad (3.16)$$

$$P_t = P_t^0 + P_t^1 H_t^1 + P_t^1 \dot{H}_t^1 \qquad (3.17)$$

In order to evaluate the dynamic coefficients of spindle journal and thrust bearings, perturbed values of :

H_j, P_j, H_t and P_j,

are substituted into the *normalised* Reynolds equation of bearing lubrication (Klit and Lund, 1986) and (Han et al 1994). Solving the normalised Reynolds using any numerical or analytical methods (mentioned earlier in the section 1.4.1), leads to the evaluation of the dynamic stiffness and damping coefficients of the spindle. For brevity, the dynamic coefficients used in this thesis to estimate the spindle run-out (ie see section 3.5) are based on data supplied by the spindle manufacturer.

3.5 ESTIMATION OF SPINDLE ERROR MOTIONS

3.5.1 INTRODUCTION

In order to understand the dynamic (mechanical) behaviour of the spindle under the influence of run-out, differential equations in the form of Langrange equations can be used (Iordanoff et al 1995) to describe the 5 DOF spindle error motion components. In this section differential equations of motion have been devised to estimate the error motions of the spindle used in this work, when it at its maximum load capacity.

3.5.2 SPINDLE DIFFERENTIAL EQUATIONS OF MOTIONS

A number of differential equations of motions have been devised to describe the 5 DOF error motions of the spindle using Lagranges's formulation [ie see general text on advanced dynamics such as Sneck (1991) for more details].
Langrange's formulation can be represented as:

$$\frac{d}{dt}\left(\frac{\delta L}{\delta \dot{n}}\right)-\frac{\delta L}{\delta n}=\Sigma Q_n \qquad (3.18)$$

Where n is the number of error motion degree of freedoms (DOFs) and L is the Lagrangian, normally given as:

$$L = T - V \qquad (3.19)$$

Where T and V are the total kinetic and potential energy values respectively. By neglecting the small potential energy contributions due to elevation changes, the Langrangian formulation given in equation 3.19 is modified to:

$$L = T \qquad (3.20)$$

hence the equation 3.18 is modified to:

$$\frac{d}{dt}\left(\frac{\delta L}{\delta \dot{n}}\right)-\frac{\delta L}{\delta n}=\dot{\mathbf{n}}[\mathbf{c}_{air}]+\mathbf{n}[\mathbf{k}_{air}] \qquad (3.21)$$

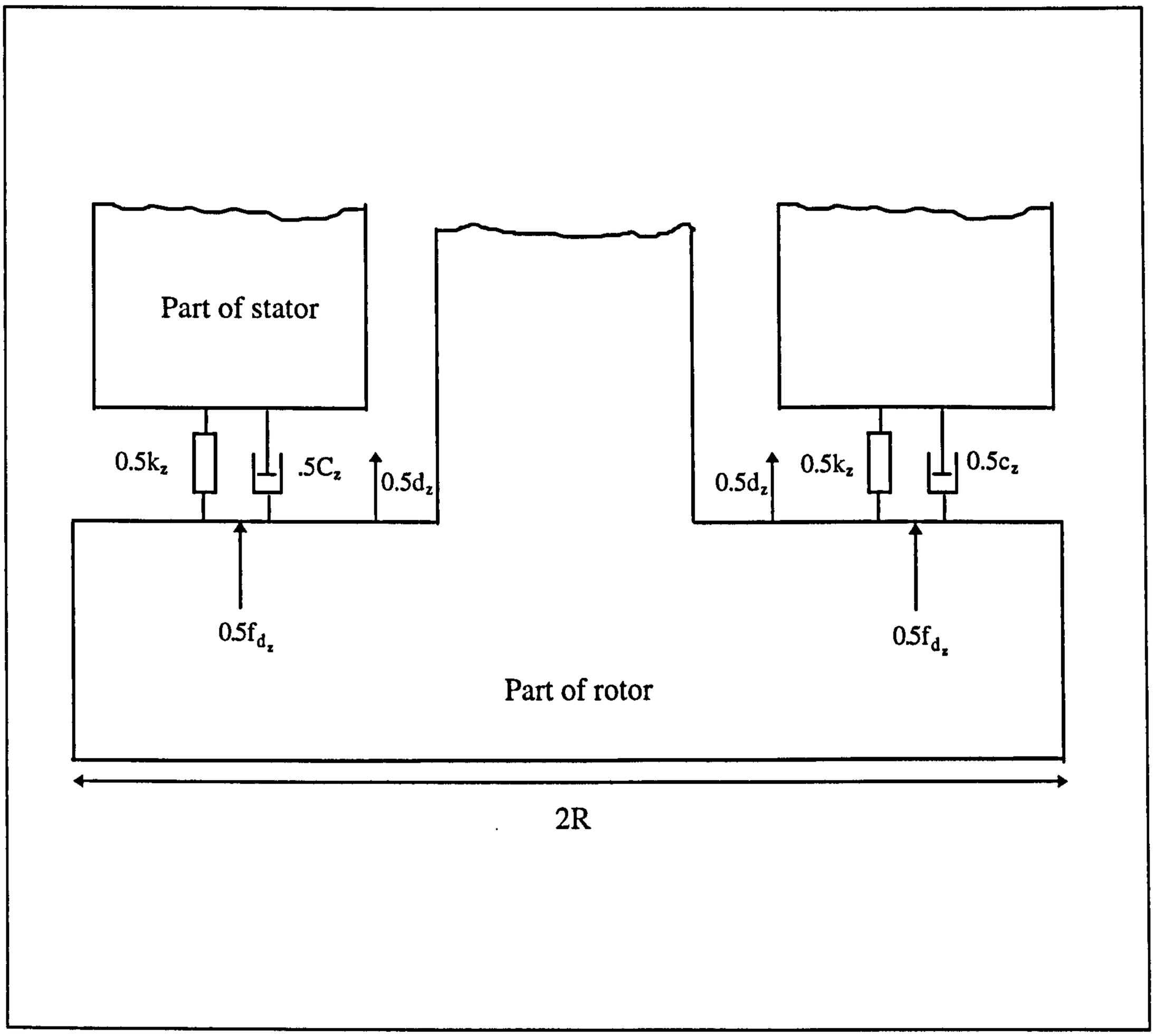


Figure 3.7a: Model used to estimate spindle axial error motions

Where **n** is any one the 5 DOF error motions given as d_x, d_y, d_z, E_x, E_y, and $\dot{\mathbf{n}}[c_{air}]$ and $\mathbf{n}[k_{air}]$ are the bearing damping and stiffness forces. According to Iordanoff et al (1995), equation 3.21 can be used to derive 5 equations:

$$m\ddot{d}_x + 2c_{d_x}\dot{d}_x + 2k_{d_x}d_x = f_{d_x}(t) \qquad (3.22)$$

$$m\ddot{d}_y + 2c_{d_y}\dot{d}_y + 2k_{d_y}d_y = f_{d_y}(t) \qquad (3.23)$$

$$m\ddot{d}_z + c_{d_z}\dot{d}_z + k_{d_z}d_z = f_{d_z}(t) \qquad (3.24)$$

$$I\dot{E}_x + 2\left(\frac{L}{2}\right)^2 c_{E_x}E_x + k_{E_x}E_x = f_{E_x}(t) \qquad (3.25)$$

$$I\dot{E}_y + 2\left(\frac{L}{2}\right)^2 c_{E_y}E_y + k_{E_y}E_y = f_{E_y}(t) \qquad (3.26)$$

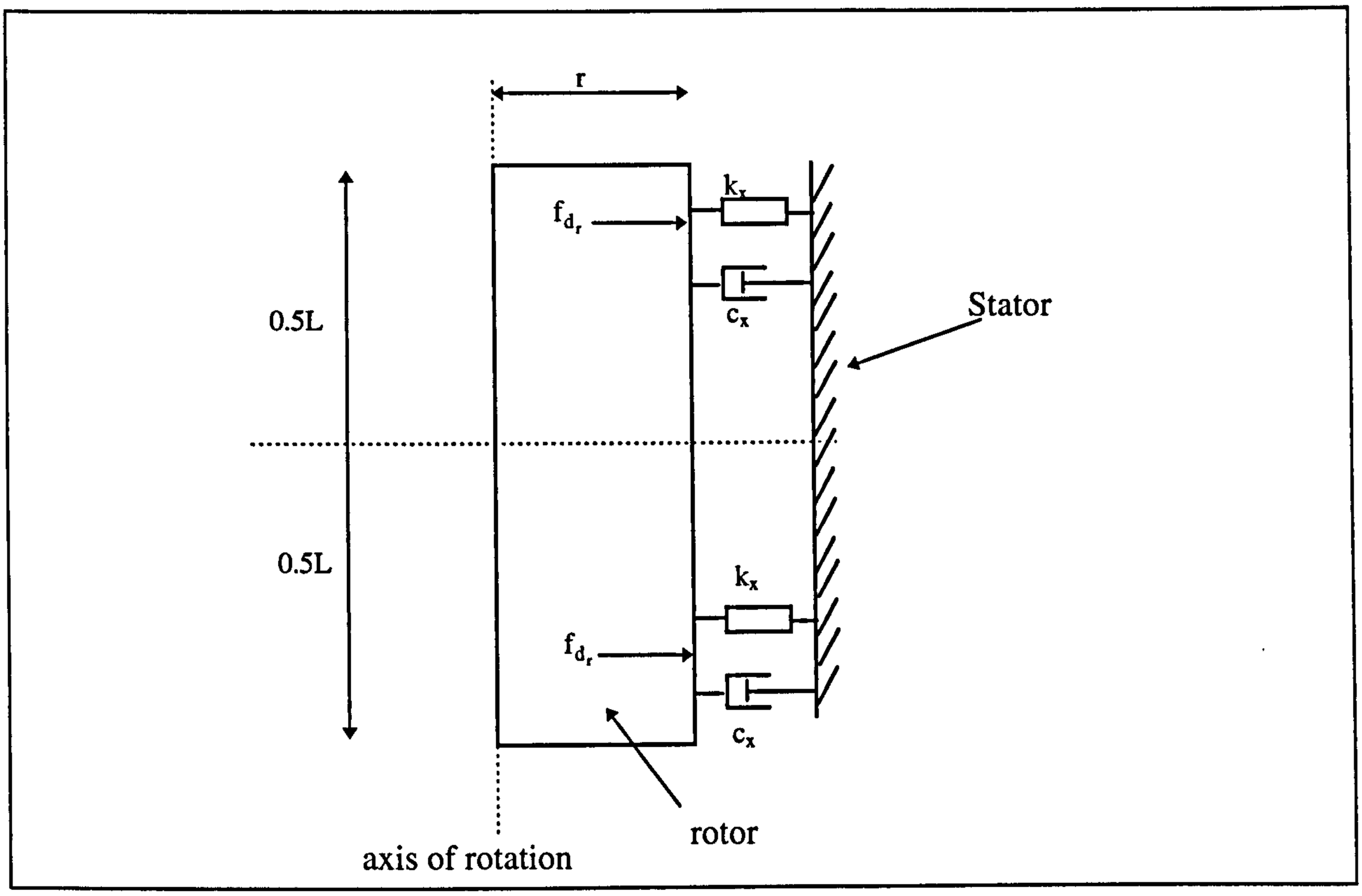


Figure 3.7b: Model used to estimate spindle radial error motions

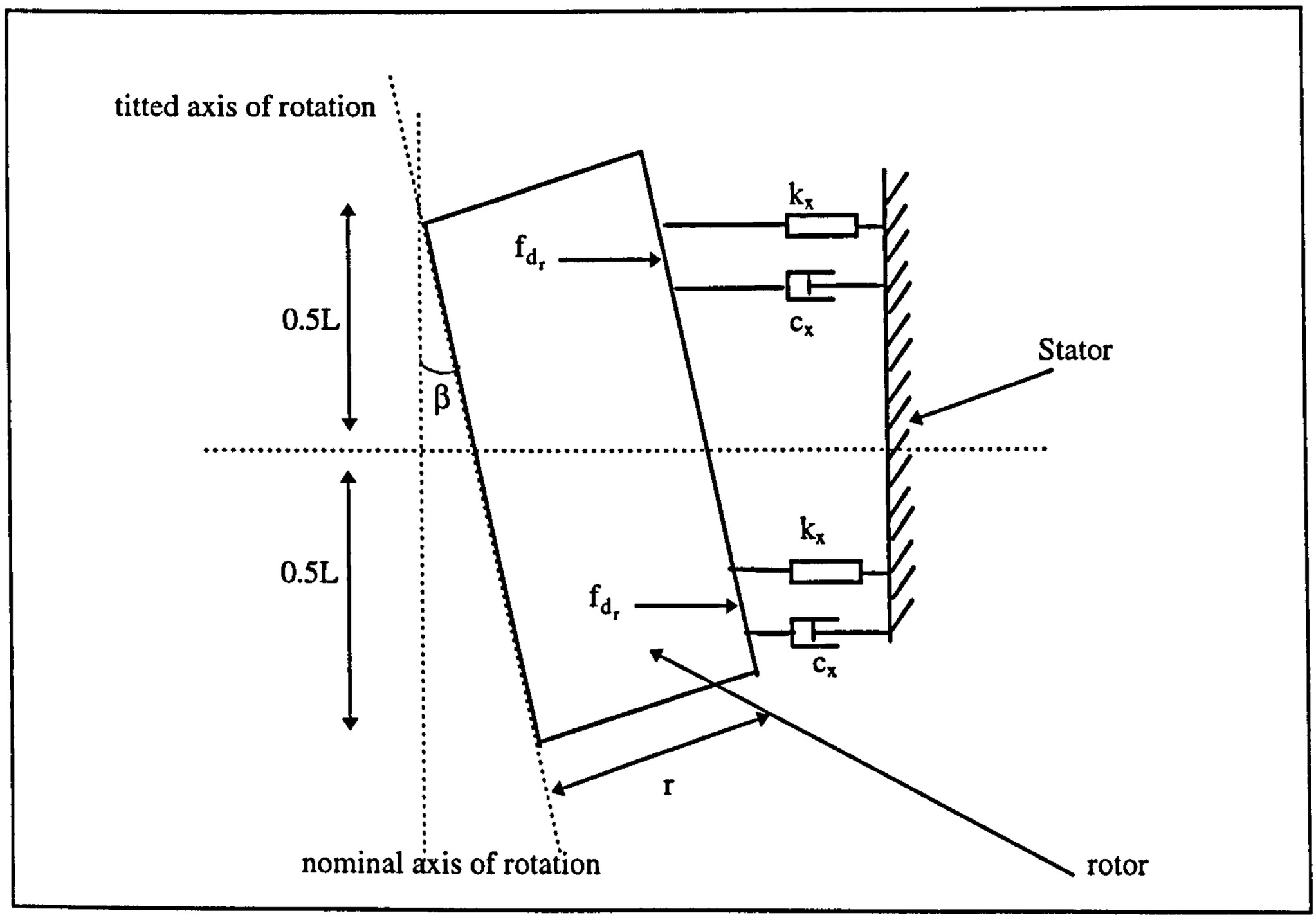


Figure 3.7c: Model used to estimate spindle tilt error motions

where,

m: mass of bearing rotor

I: Inertia of rotor

$c_{d_x} = c_{d_y}$: damping coefficient of bearing in the radial direction

c_{d_z}: damping coefficient of bearing in the axial direction

$c_{E_x} = c_{E_y}$: damping coefficient of bearing in the angular direction

$k_{d_x} = k_{d_y}$: stiffness coefficient of bearing in the radial direction

k_{d_z}: stiffness coefficient of bearing in the axial direction

$f_{d_x} = f_{d_y}$: load capacity of bearing in the radial direction

f_{d_z}: load capacity of bearing in the axial direction

$f_{E_x} = f_{E_y}$: load capacity of bearing in the angular direction

Equations 3.22 to 3.26 indicate that spindle error motion is influenced by inertial, damping and elastic forces. Figures 3.7a to 3.7c show the schematic representation of these forces with the aid of mass, spring and damping elements.

3.5.3 ESTIMATED ERROR MOTIONS OF THE SPINDLE

An estimation of the 5 DOF error motions of the spindle used in this work is given in this section. This estimation is based on the Laplace transform of equations 3.22 to 3.26 and for brevity ($E_t = E_x = E_y$) and ($d_r = d_x = d_y$), this simplifies the equations to:

$$ms^2D_r + 2c_{d_r}D_r s + 2k_{d_r}D_r = f_{d_r}(s) \quad (3.27)$$

$$ms^2D_z + c_{d_z}D_z s + k_{d_z}D_z = f_{d_z}(s) \quad (3.28)$$

$$Is^2\beta + 2\left(\frac{L}{2}\right)^2 c_{d_t}\beta s + 2\left(\frac{L}{2}\right)^2 k_{d_t}\beta = f_{d_t}(s) \quad (3.29)$$

where,

D_r: d_r in Laplace domain

D_z: D_z in Laplace domain

β: E_t in Laplace domain

$m = 3.5$ kg
$I = 5.16 \times 10^{-4}$ Nm^2
$f_{d_r} = 445$ N
$f_{d_z} = 1780$ N
$f_{d_t} = 45$ N
$k_{d_r} = 117 \times 10^6$ N/m
$k_{d_z} = 350 \times 10^6$ N/m
$k_{d_t} = 45 \times 10^6$ Nm/μrad
$\alpha = 1.411$ (ratio of heat capacities of bearing air supply)
$P_s = 1034$ kPa (bearing air supply pressure)
$\dot{m} = 9.5 \times 10^{-4}$ m^3 / min (bearing air volume flow rate)
$T = 293$ K (approximate temperature of bearing during operation)
$R = 293$ J/mol K (universal gas constant)
$d = 1 \times 10^{-3}$ m (diameter of bearing supply feed slots)
$h = 500 \times 10^{-6}$ m (bearing clearance)
$B = 0.1$ m (width of bearing rotor)
$L = 0.1$ m (length of bearing rotor)
$N = 10{,}000$ rev/min (maximum speed of bearing)
$r = (L/2) = 0.05$ m (radius of bearing rotor)

Table 3.1: Bearing data supplied by manufacturer

The error motions of the spindle are evaluated using bearing data given in table 3.1 (provided by spindle manufacturer) and the estimated damping coefficient data given in appendix 1. Substituting the values of the mass, inertia, stiffness and damping coefficients, and the bearing load capacity values into equations 3.27 to 3.39 give:

$$D_r(s) = \frac{445}{3.5s^2 + 4.581 \times 10^{10} s + 234 \times 10^6} \tag{3.30}$$

$$D_z(s) = \frac{1780}{3.5s^2 + 9.163 \times 10^{10} s + 350 \times 10^6} \tag{3.31}$$

$$D_t(s) = \frac{45}{5.16 \times 10^{-4} s^2 + 1.145 \times 10^8 s + 2.1 \times 10^3} \tag{3.32}$$

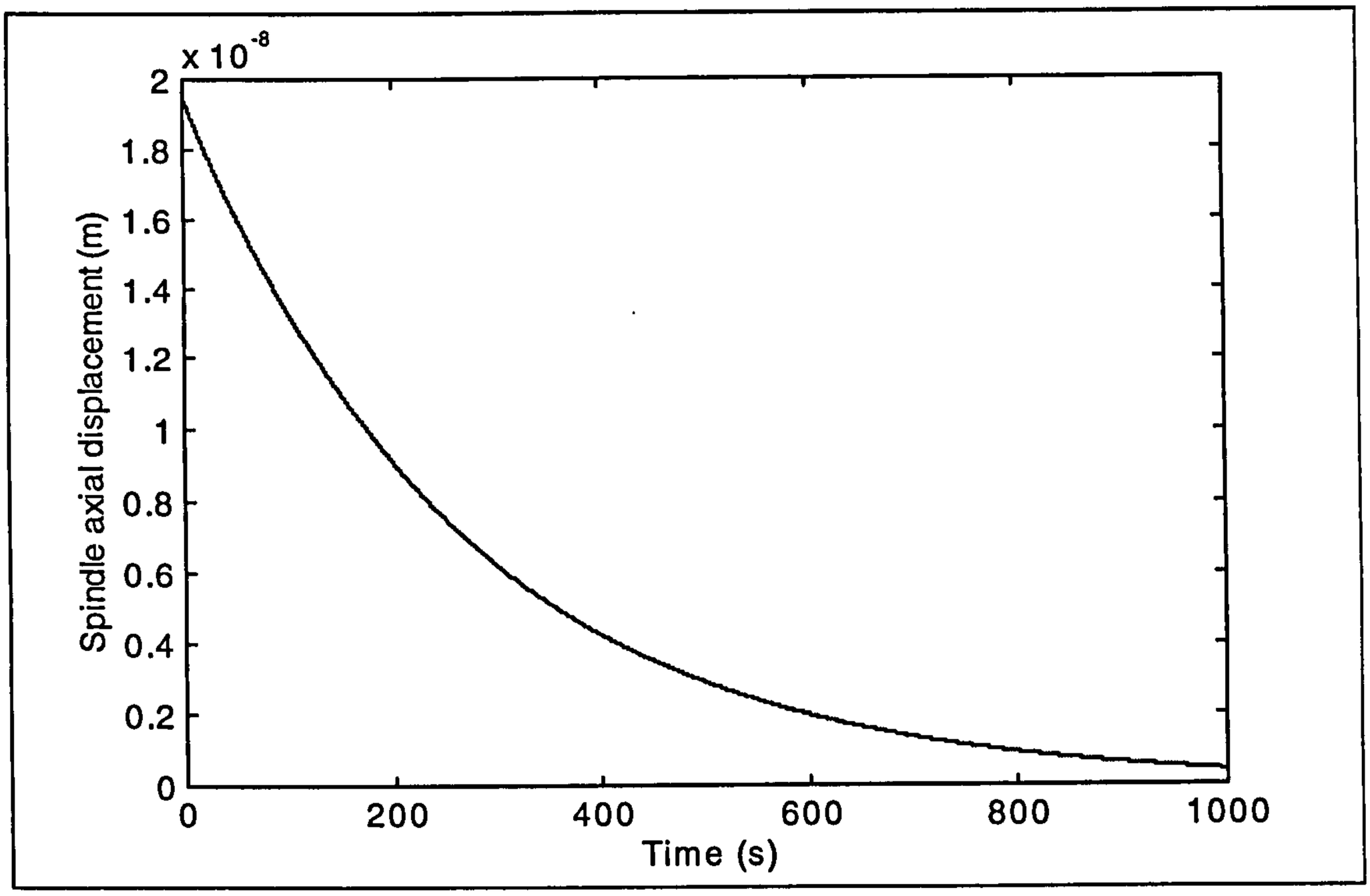


Figure 3.8a: Transient response of the estimated axial error

Hence, the error motion values can be formulated in the time domain by the inverse Laplace transforms of equations 3.30 to 3.32:

$$d_r(t) = 9.71\times10^{-9}\left(e^{-5.1\times10^{-3}t} - e^{-1.31\times10^{10}t}\right) \quad (3.33)$$

$$d_z(t) = 19.4\times10^{-9}\left(e^{-3.8\times10^{-3}t} - e^{-2.61\times10^{10}t}\right) \quad (3.34)$$

$$d_t(t) = 0.19\times10^{-6}\left(e^{-1.84\times10^{-5}t} - e^{-2.221\times10^{11}t}\right) \quad (3.35)$$

Figures 3.8a to 3.8c show the transient response of the spindle as defined in equations 3.33 to 3.35. The figures indicate that the *peak* (angular, axial and radial) displacement of the spindle occurs at time (t) = 0. At t > 0, the displacement exponentially approaches zero.

Estimated values of bearing radial, axial and tilt error motions are *peak* displacements 9.7 nm, 19.4 nm and 0.19 μrad respectively, for a bearing operating at 1 MPa (as shown in figures 3.33, 3.34 and 3.35). There is a deviation between these values and those *stated* by the bearing manufacturer (Slocum, 1992, pp623- 624) as shown in table 3.3. This is due to the assumptions made in the error motion estimation, such as:

- bearing has *static* stiffness (this is untrue in reality since the pressure gradient in bearing clearance varies during spindle operation causing it to have *dynamic* stiffness)

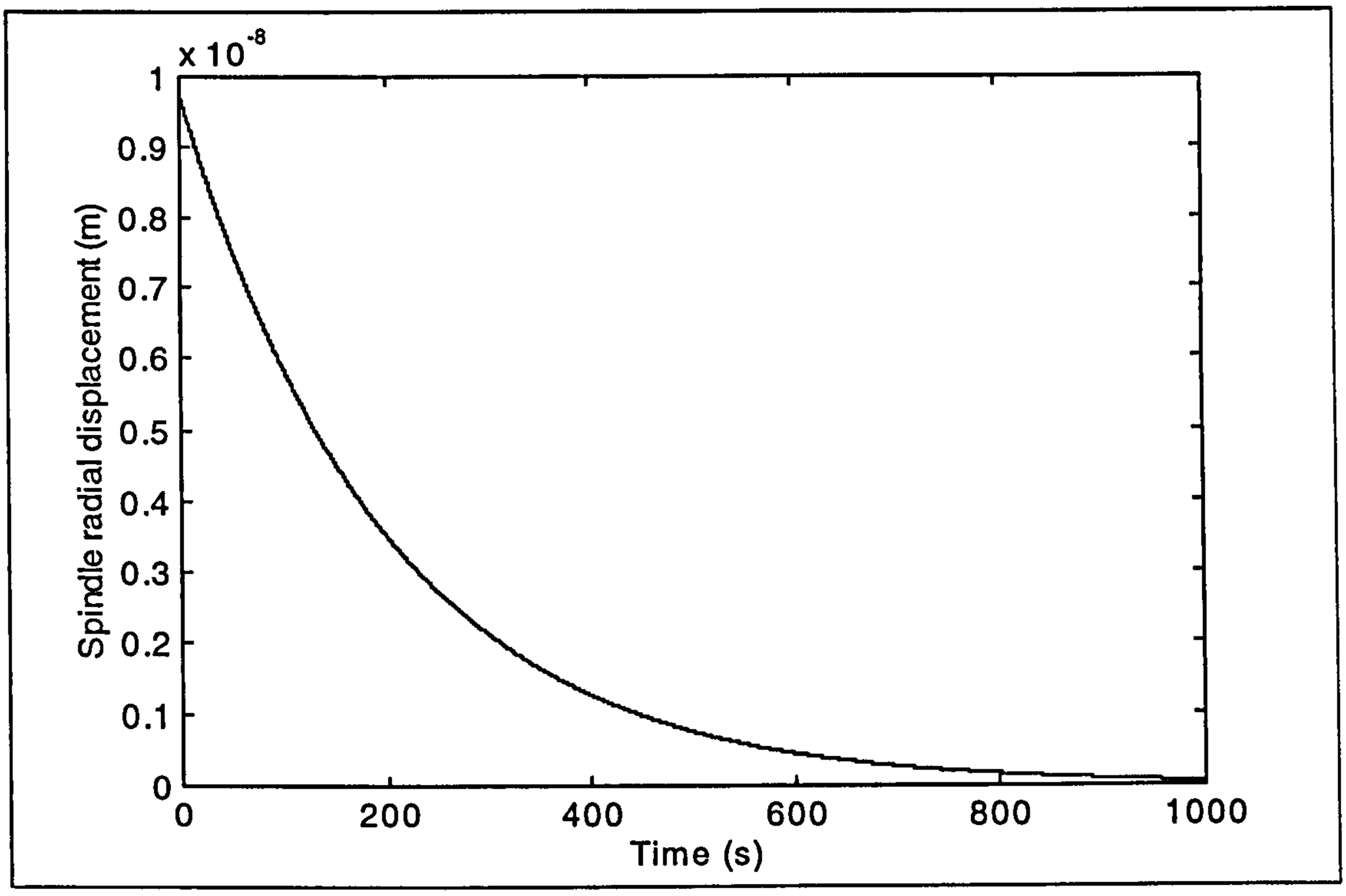


Figure 3.8b: Transient response of the estimated radial error

Type of error motion	*Estimated* data obtained	Manufacturer's data (modified)	Deviation
Axial	19.4 nm	< 50 nm	(with spec.)
Pitch	9.71 nm	< 50 nm	(with spec.)
Roll	0.19 μrads	< 0.20 μrads	(with spec.)

Table 3.2: Estimated and stated error motions of bearing (operating at 1 MPa)

- damping coefficients of bearing in the angular and radial directions are equal (see table 3.2)

- damping and stiffness coefficients in the radial direction are equal at both ends of the rotor

- maximum forces acting on the bearing during operation are equal to the their maximum load capacity.

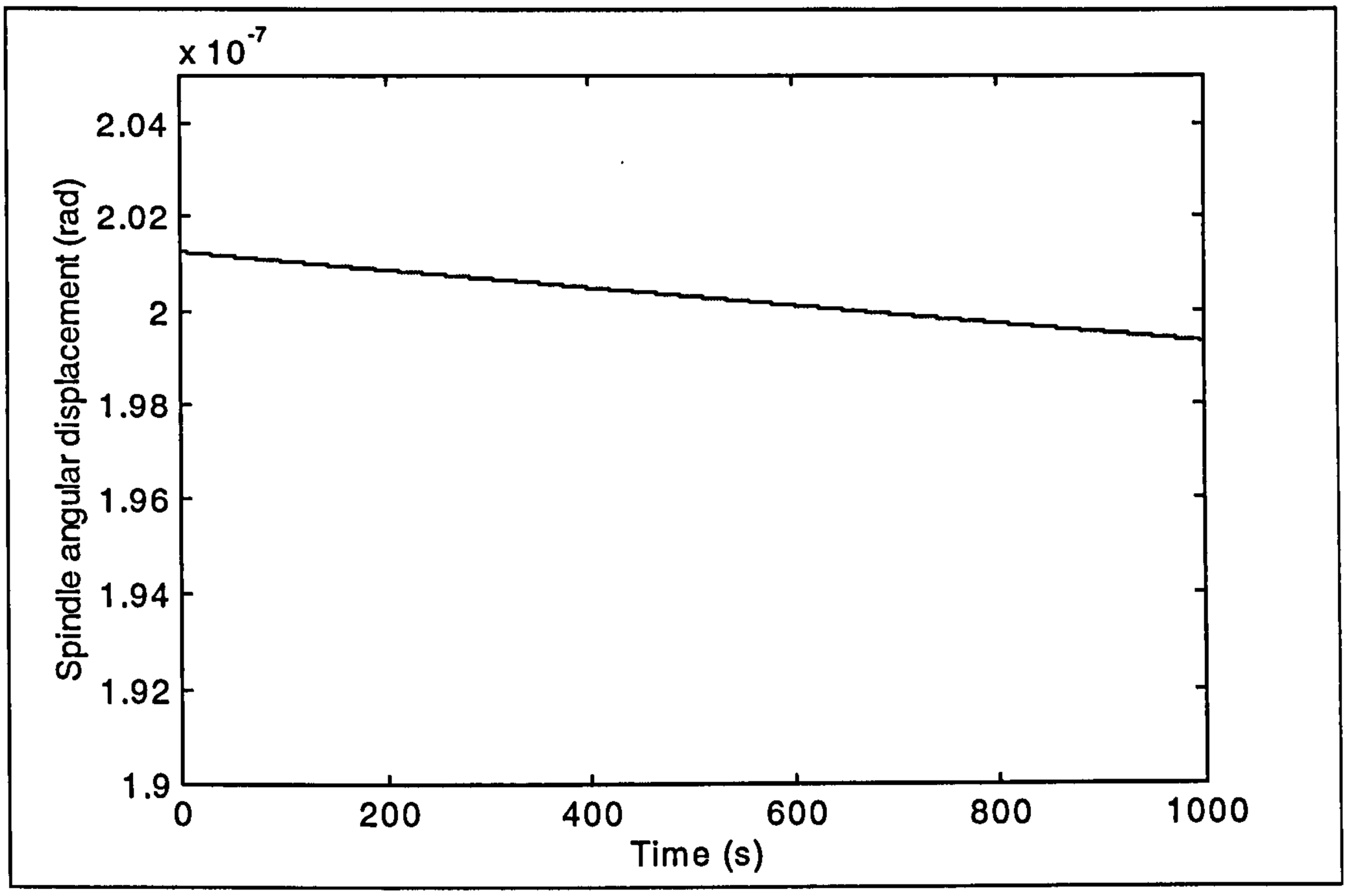


Figure 3.8c: Transient response of the estimated tilt error

3.6 CAUSES OF SPINDLE ERROR MOTIONS

3.6.1 INTRODUCTION

In this section, a description is given on the causes of error motions in most gas bearings. According to the ASME/ANSI American National Standard (1985), the principal causes of spindle error motions are:

- Bearing geometric form errors
- Structural error motions
- Thermal effects

3.6.2 BEARING GEOMETRIC FORM ERRORS

Generally bearing geometric form errors are due to the form imperfections of the spindle rotor, stator and restrictor components. These imperfections are:

- form errors of bearing rotor
- surface roughness errors
-

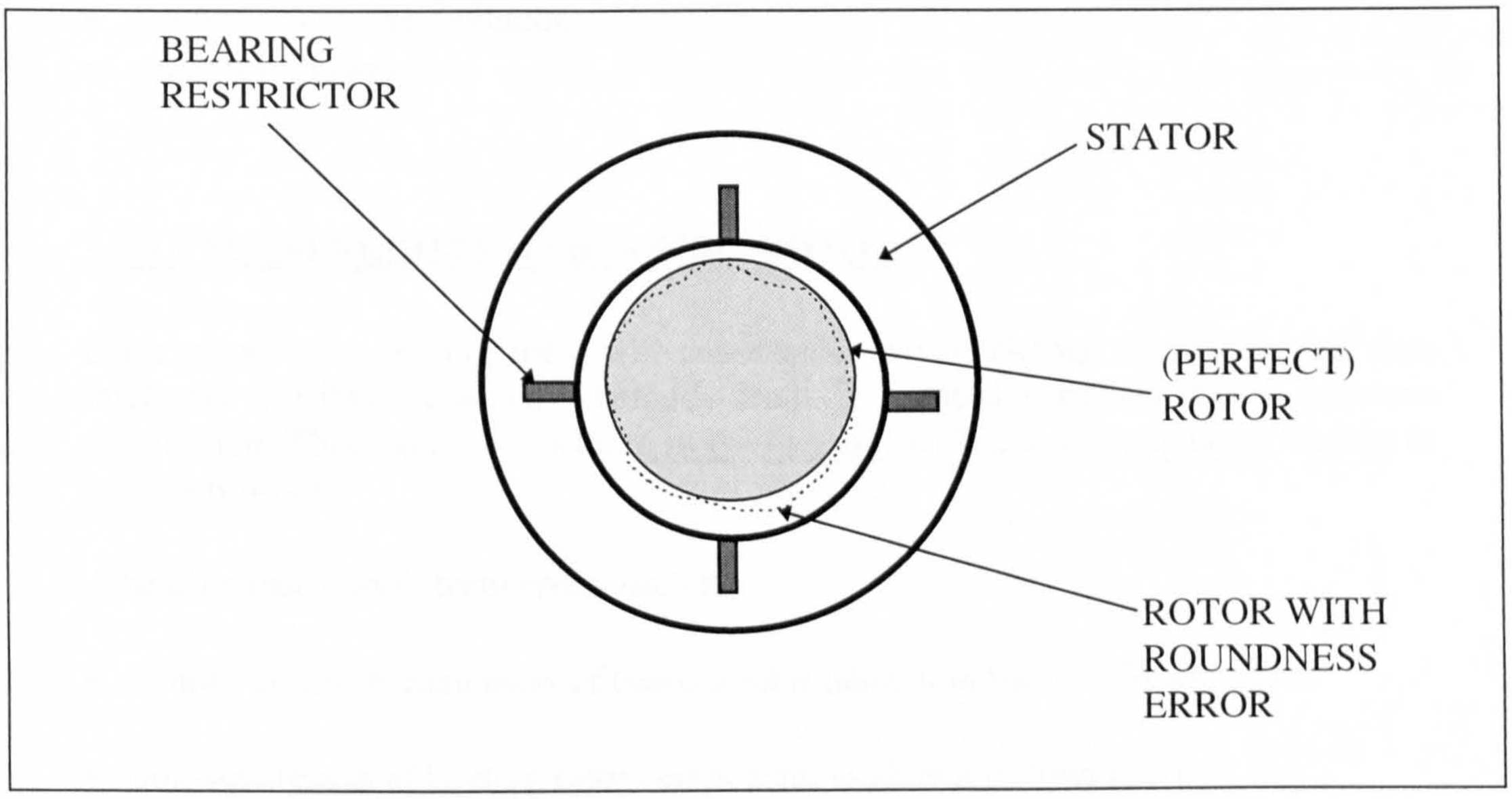


Figure 3.9: Roundness error of the spindle rotor

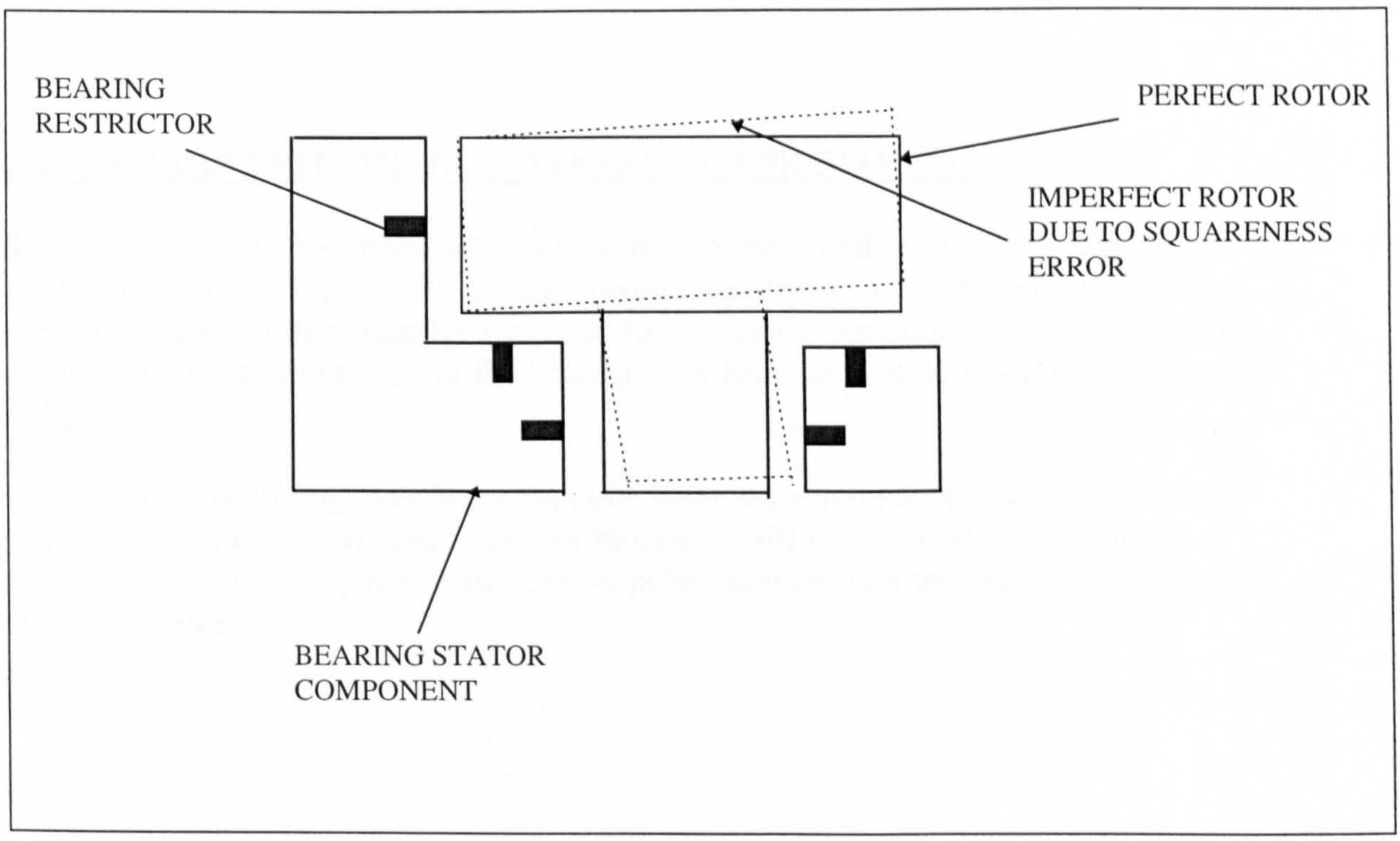


Figure 3.10 : Squareness errors of spindle rotor

- form errors of bearing stator
- restrictor hole size deviation
- restrictor blockage

3.6.2.1 FORM ERRORS OF BEARING ROTOR

Form errors of a bearing rotor will cause an invariant change in bearing gas film thickness distribution which inherently leads to a rapid variation in its pressure distribution. This causes a reduction in the bearings stiffness, consequently leading to spindle run-out.

A bearing can exhibit form errors such as:

- ovality or out-of-roundness of bearing rotor (shown in figure 3.9) and stator
- non-squareness of bearing rotor component, as shown in figure 3.10.
- surface roughness and planarity errors associated with the aerostatic spindle components i.e. rotor, stator, restrictors, grooves and slots.

3.6.2.2 ERROR MOTIONS DUE TO RESTRICTOR BLOCKAGE

Spindle run-out in the from of radial error motions, will occur as a result of a malfunction or blockage of one or more restrictor *pockets*. This will cause the pocket pressure at the outflow (faulty) restrictor to rise below its nominal pressure. Thus causing an overall imbalance of the bearing rotor from its centred position, as shown in figure 3.11.

One way of quantifying the effect of spindle radial error motions (sometimes referred to as *whirl* or *precession)* due to pocket blockage, will be to correlate the number of *lobes* observed in the spindle error-motion polar diagram with the number of bearing restrictor pockets.

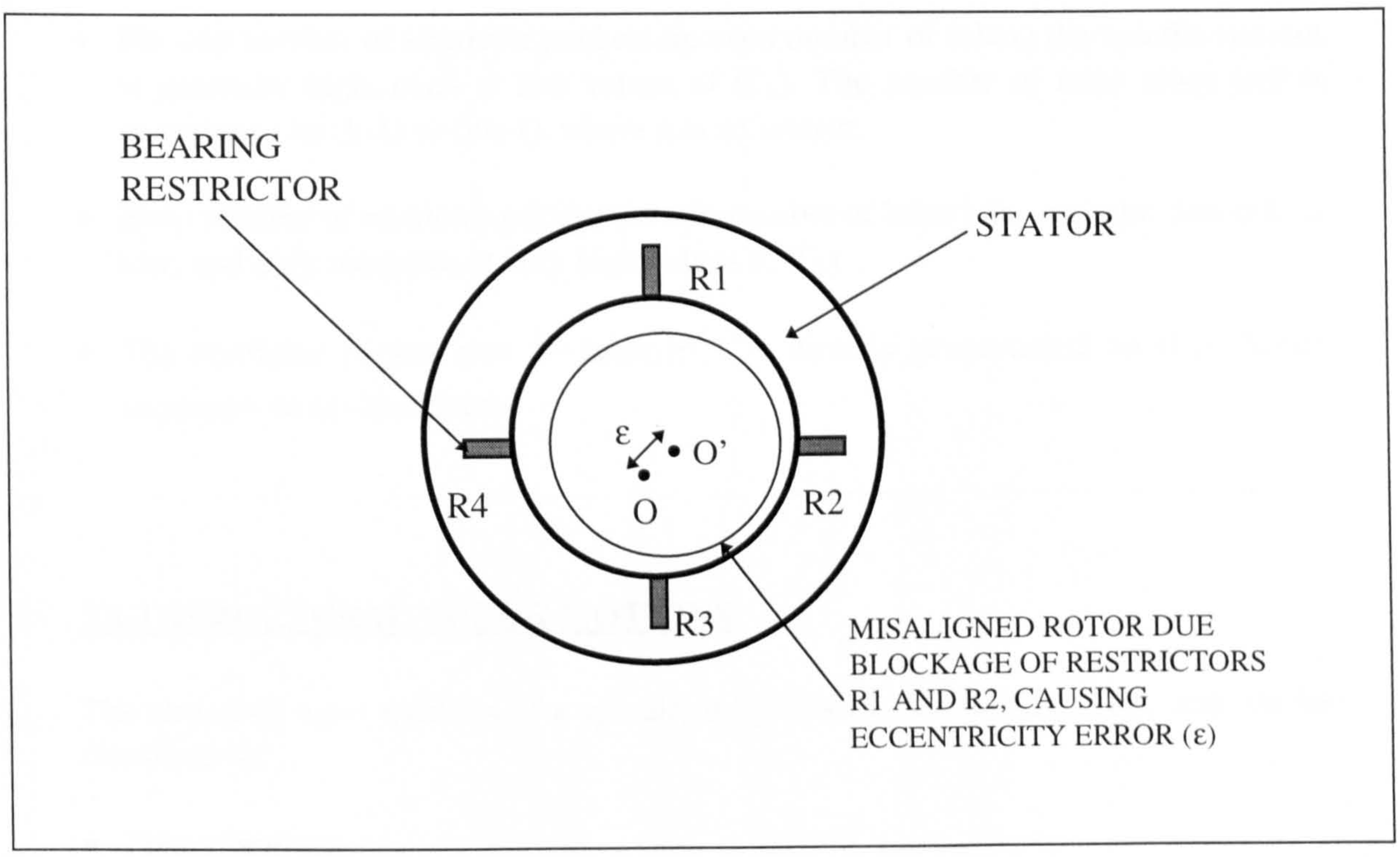


Figure 3.11 : Eccentricity error due bearing pocket blockage

3.6.2.3 ERROR MOTIONS DUE TO ROTOR AND POCKETS FORM ERRORS

Spindle rotational accuracy will be affected by the number of lobes (m) present on a rotor, and accuracy of bearing restrictor pockets. Yabe (1994a & 1994b) investigated the effect of rotor and restrictor pocket size deviation, on the performance of aerostatic bearings. This work involved the analysis of the both static and dynamic spindle run-out (using a numerical approach) . It also investigated how optimal designs of an aerostatic spindle can be achieved by proper understanding of bearing parameters such as:

- restrictor hole size (r'_s)
- number of restrictor holes (k)
- deviation of cross section of the rotor (δ),
- bearing *feeding parameter* (Γ_k), (ie this parameter relates to the stiffness and load carrying capacity of the bearing as described in section 2.5.4)

A summary of Yabe's (1994a, 1994b) findings are:

- During rotation the spindle run-out (ε) would increases proportionally with (Γ_k)

- For odd number of restrictor pockets (ie even number of lobes) the spindle run-out is generally high, even at low values of (Γ_k). The number of rotor lobes (m) is assumed to be (k-1) or (nk-1), where n is an integer.

- Even number of restrictor pockets (ie odd number of lobes) the spindle run-out is low, and only increases at very high values of (Γ_k) .

- The restrictor pocket size deviation (r'_s) is directly proportional to (Γ_k), hence increases, as (ε) increases.

3.6.3 STRUCTURAL ERROR MOTIONS

The structural error motions of a spindle are primarily due to vibrations, and can be classified as:

- Free vibrations
- Forced vibrations
- Self-induced vibrations

3.6.3.1 FREE VIBRATIONS

Free vibrations are due to random fluctuations of an unloaded spindle (ie it has no external load apart from its weight) as it rotates. Spindle free vibrations occur as a result of the moderate-to-low damping capabilities of a bearing in its normal and tangential bearing directions, (Slocum, 1992 pp580 - 625).

In order to suppress free vibrations of a machine tool bearing, the compliance of all the associated components within the machine structural loop (spindle, workpiece, fixture, tool, probes) must ideally be zero and kept constant (ANSI/ASME B89.3.4M, 1985). Reducing the compliance of the bearing in some cases requires the use of *active film restrictors*. Generally, active restrictors use either an air film-thickness feedback method or a bearing-pressure feedback method. In the former method of active restriction, the change in bearing air film thickness is detected electrically and piezo-actuators are used for controlling the film restriction. In the latter method, the change in pressure in bearing air film controls the opening of the active restrictor.

Mizumoto et al (1990) developed a zero-compliance aerostatic spindle thrust bearing that incorporated a pressure feedback active film restrictor, which was referred to as an aerostatically-controlled restrictor (ACR). The moving member in the ACR was supported elastically by another aerostatic bearing used within the spindle thrust bearing. Experiments show that the compliance of the thrust bearing with the ACR can be less than 1 nm N^{-1} when the frequency of the vibratory disturbance is less than 2 Hz.

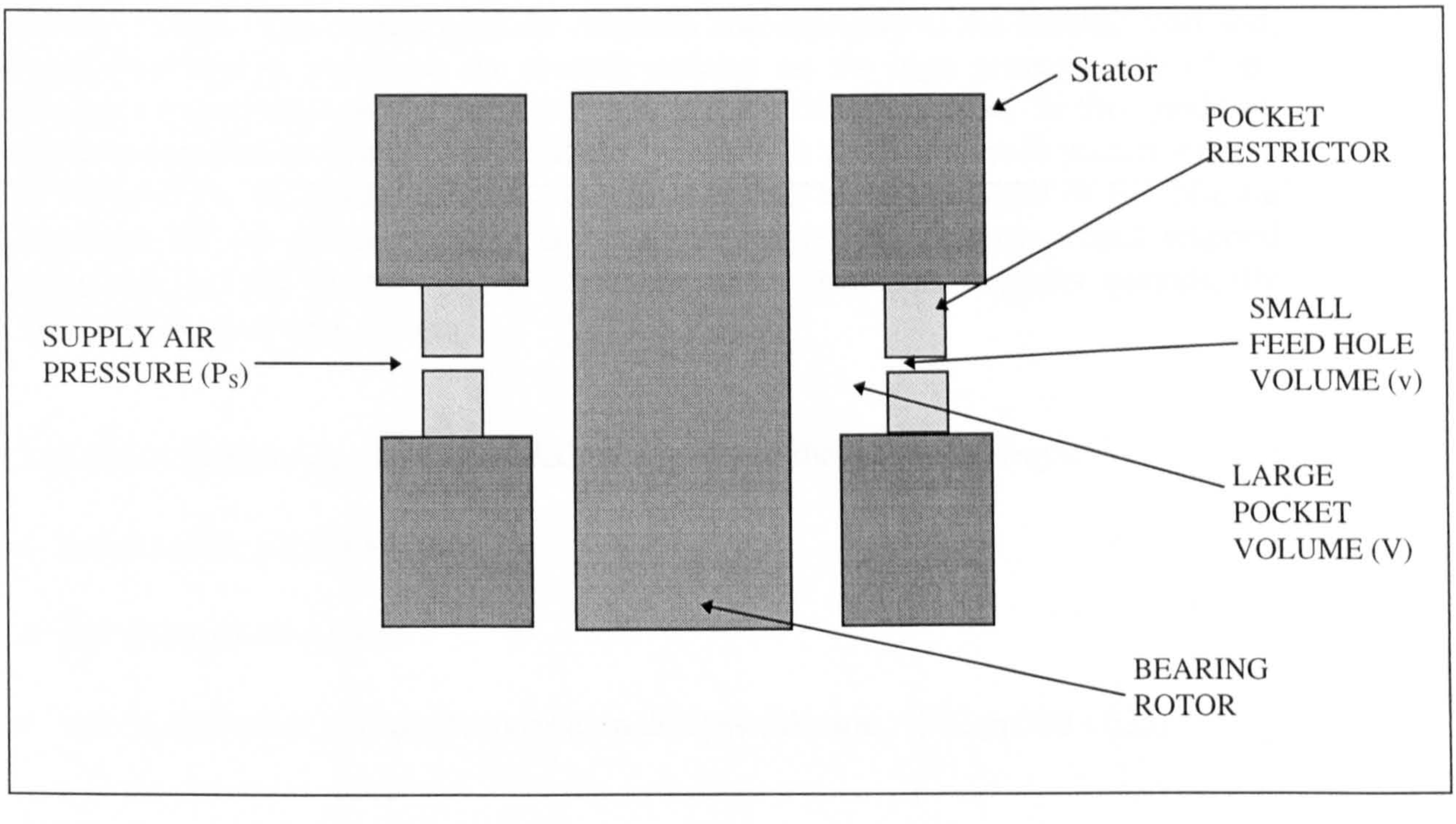


Figure 3.12: Journal bearing showing pocket restriction

The main cause of free vibrations in high-speed aerostatic bearings is pneumatic instability such as *self-excited resonance.* Self-excited resonance is normally associated with choked or nearly choked feed hole conditions and the frequency of vibration approaching the natural frequency of the rotor mass and bearing stiffness (Grassam & Powell, 1964). This type of instability is excited by random pressure fluctuations, associated with shock waves in the bearing, which cause the bearing to vibrate at its natural frequency. Self-excited resonance can be avoided by designing the bearing to incorporate smaller clearances and increased static bearing stiffness.

3.6.3.2 FORCED VIBRATIONS

Forced vibrations are the periodical fluctuating forces arising from a bearing: such as *pneumatic hammering*, *whirl* and *pulsing effect of spindle torque transmission.* Pneumatic hammering occurs when the bearing components of the spindle resonate as the air film alternately compresses and expands periodically (Slocum, 1992 pp 580-625). This phenomenon is caused by the compressibility of gases and the consequent delay between bearing clearance changes and the response to this change through

variations in pressure in the bearing restrictor. In journal bearings the instability may be in the form of radial vibration of the shaft, while in thrust bearings the instability can produce axial vibration of the rotor, which may or not be audible (Grassam & Powell, 1964).

Pneumatic hammering occurs when the gas is fed through a small feed hole volume (v) into a pocket of relative large volume (V), as shown in figure 3.12 (Grassam & Powell, 1964). The pocket pressure responds automatically to the bearing load and, when this load is increased, the journal surfaces on the high pressure side of the bearing move closer together while the flow in the feed hole reduces. In this condition the pocket pressure approaches the supply pressure. A small change in pocket volume is required for the pocket pressure to adjust rapidly to any changes in the bearing clearance. If the pocket volume is relatively large, the bearing would respond sluggishly to any changes in its clearance and, when this happens periodically pneumatic hammering occurs.

Pneumatic hammering can be avoided by any one of the following ways:

- reducing the pocket volume
- use of larger feed holes
- use of alternative orifice compensation designs (Slocum, 1992 pp580 - 625)

Whirl is another source of forced vibrations encountered in high-speed spindles, and can be defined as a relative orbital motion of bearing rotor with respect to its stator. There are two types of bearing whirl, these are *synchronous whirl* and *fractional speed whirl* (Grassam & Powell, 1964). Synchronous whirl is generally due to rotor unbalance, it occurs at all speeds of the rotor but is most critical when the spindle speed is near or at one of the rotor natural frequencies (Sneck, 1991). Fractional speed whirl on the other hand causes rotor unbalance as a result of the occurrence of resonance at rotational frequencies equal to the fractional speeds. The most common type of fractional speed whirl is known as *half-frequency whirl*, and occurs when the natural frequency of the bearing is equal to half the rotational frequency.

According to Rieger (1970), whirl is caused by residual rotor unbalance due to a small radial displacement of the centre of gravity (CG) of the rotor from its geometric axis. The CG is driven further outward from the axis of rotation by the centrifugal force acting on it through rotation, so that the speed of rotation and the whirling motion of the rotor are synchronous. Whirl is generally stable at all speeds except at the *critical* speeds of the machine. Balancing the rotor and operating away from the critical speeds adequately minimises the size of the whirl orbit.

Small whirl components often exist in the spindle motion at all times, coexisting with the main half-frequency whirl component. In this condition, the rotor has a natural tendency to *pump* the gas around the bearing clearance, causing rotor imbalance. However, this occurrence is usually suppressed by the damping present in the film

until the speed of the spindle approaches its lowest value of critical speed (N_c). In this condition the bearing damping falls to zero and the rotor whirl amplitude (radial error motion) will grow rapidly within the bearing clearance with respect to its centred position.

Another example of forced vibrations that might affect the spindle accuracy is the *pulsing effect* of a bearing drive torque transmission mechanism, such as a vee belt. This can be avoided by incorporating a *direct drive* motor to the spindle.

3.6.3.3 SELF - INDUCED VIBRATIONS

Self-induced vibrations are generated during a machining or measuring process usually due to the fluctuating effect of a tool or probe on the spindle of a machine tool or measurement instrument respectively.

3.6.3.4 REDUCING VIBRATION IN SPINDLES

Vibratory motion is a major contribution to spindle rotational inaccuracy, so it is important to reduce bearing vibrational effects. The reduction of spindle vibratory motion requires:

- amplitude of the error motion must be very small compared with the bearing clearance, so, ideally, the bearing clearance should be as small as possible
- operating the spindle at speeds much lower than the smallest critical speed of the bearing.
- improving damping characteristics of bearing so that it can absorb the vibratory energy (Oshumi et al, 1984).
- infinite stiffness of the spindle bearing and its components within its structural loop.

3.6.4 INFLUENCE OF THERMAL EFFECTS ON SPINDLE ACCURACY

3.6.4.1 INTRODUCTION

Thermal effects can contribute to 50% of the overall error in machine tool performance (Mckeown et al, 1995). Bryan (1968) emphasised the necessity of understanding the influence of thermal errors in machine tools and metrology at large. Heat sources that cause thermal drifts in aerostatic spindles can be classified as *internal* and *external* (see

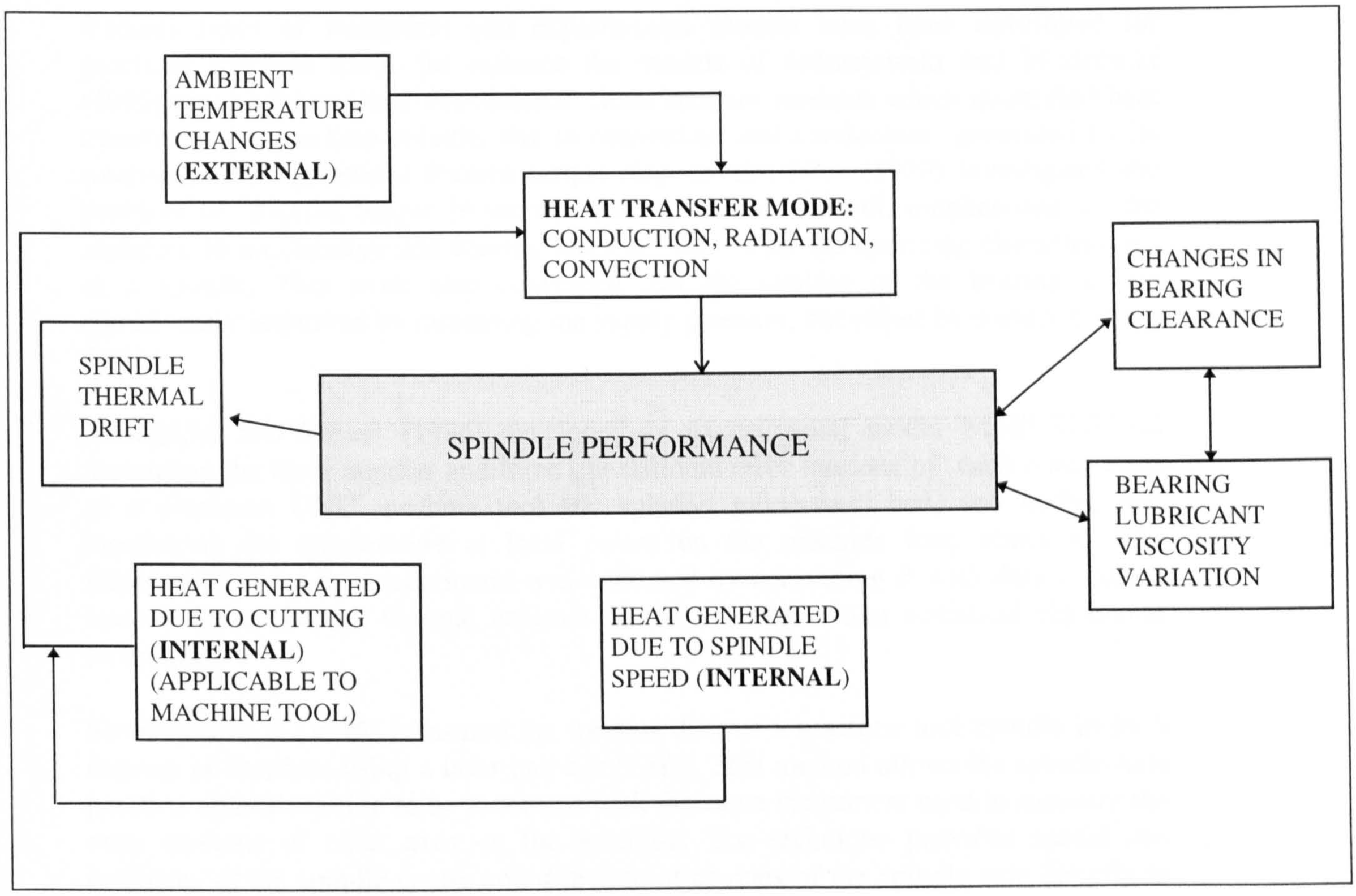


Figure 3.13: Classification of heat sources in spindles

figure 3.13). Internal heat sources are due to convective and conductive heat transfer within the bearing viscous film and its journal/thrust surfaces. External heat sources, on the other hand, are due to any ambient temperature variations causing temperature gradients which in turn cause thermo-elastic deformation of the spindle.

In this section, a brief review of past research on thermal effects on spindle performance is given. Also a brief description of heat generation and thermal growth in air bearings is presented, including an outline on possible ways to reduce them.

3.6.4.2 SOME PAST RESEARCH ON THERMAL EFFECTS IN SPINDLE BEARINGS

A number of workers have studied the influence of thermal effects on spindle accuracy using *thermal modelling*. Thermal modelling can be categorised as *numerical* and *experimental* techniques where the former involves application of finite element or finite difference methods for solving the energy and/or momentum equations and the latter involves direct measurement of the temperature gradient of a rotating spindle (Mckeown et al, 1995). Experimental modelling of thermal effects in machine tools allows for the provision of direct compensation of thermal drifts.

Various types of numerical and experimental models have been developed for precision machine tools, for instance the models of Jedrzejewski and Modrzycki (1992) was based on three-dimensional finite element methods which quantified heat transfer in the machine spindle, due to convection and conduction generated by its rotation and its generated friction torque respectively. Nica (1970) investigated the problem of friction torque in journal bearings in more detail, emphasising on the variation in temperature and bearing lubricant flow with the operating characteristics of a spindle. This work also confirmed that the cooling of the bearing is not significantly improved by increasing the supply pressure, but rather by correct bearing design.

Venugopal and Barash (1986) developed an experimental model which involved measuring the three angular and three translational error motions of each component of a precision CNC machine tool (its spindle, guide-way, bed, and body), and monitoring the temperature at local points on the machine loop structure. The information from this experiment was validated by correlating it with data acquired from a finite element thermal analysis of the machine, using statistical regression techniques.

Shrinivasa et al (1996) measured the thermal drift of a machine tool spindle in its 5 degrees of freedom using a laser ball bar (LSB). This method allows the spindle axis position and orientation to be measured with the same instrument used to measure the error motions of other axes on the machine. The technique provides spatial co-ordinates of the spindle centre and direction of cosines of the spindle axis directly in machine co-ordinates.

Baker and Hornung (1970) gave a comprehensive heat transfer analysis for an aerostatic bearing, considering the effects of viscosity variation and thermal effects due to the net heat generated in the spindle shaft and housing, and effects of bearing lubricant (air).

3.6.4.3 HEAT GENERATION AND THERMAL GROWTH IN AIR BEARINGS

According to Baker and Hornung (1970), thermal effects in air bearings are primarily due to:

- heat generated (q_g) due to spindle torque (Γ)
- heat transferred (q_a) by convection of air to bearing fluid film
- heat transferred (q_r) by conduction to bearing rotor
- heat transferred (q_s) by conduction/convection to bearing stator (housing)

The heat generated q_g is dependent on spindle speed (ω) and is the sum of the other terms:

$$q_g = q_a + q_s + q_r = \Gamma\omega \qquad (3.36)$$

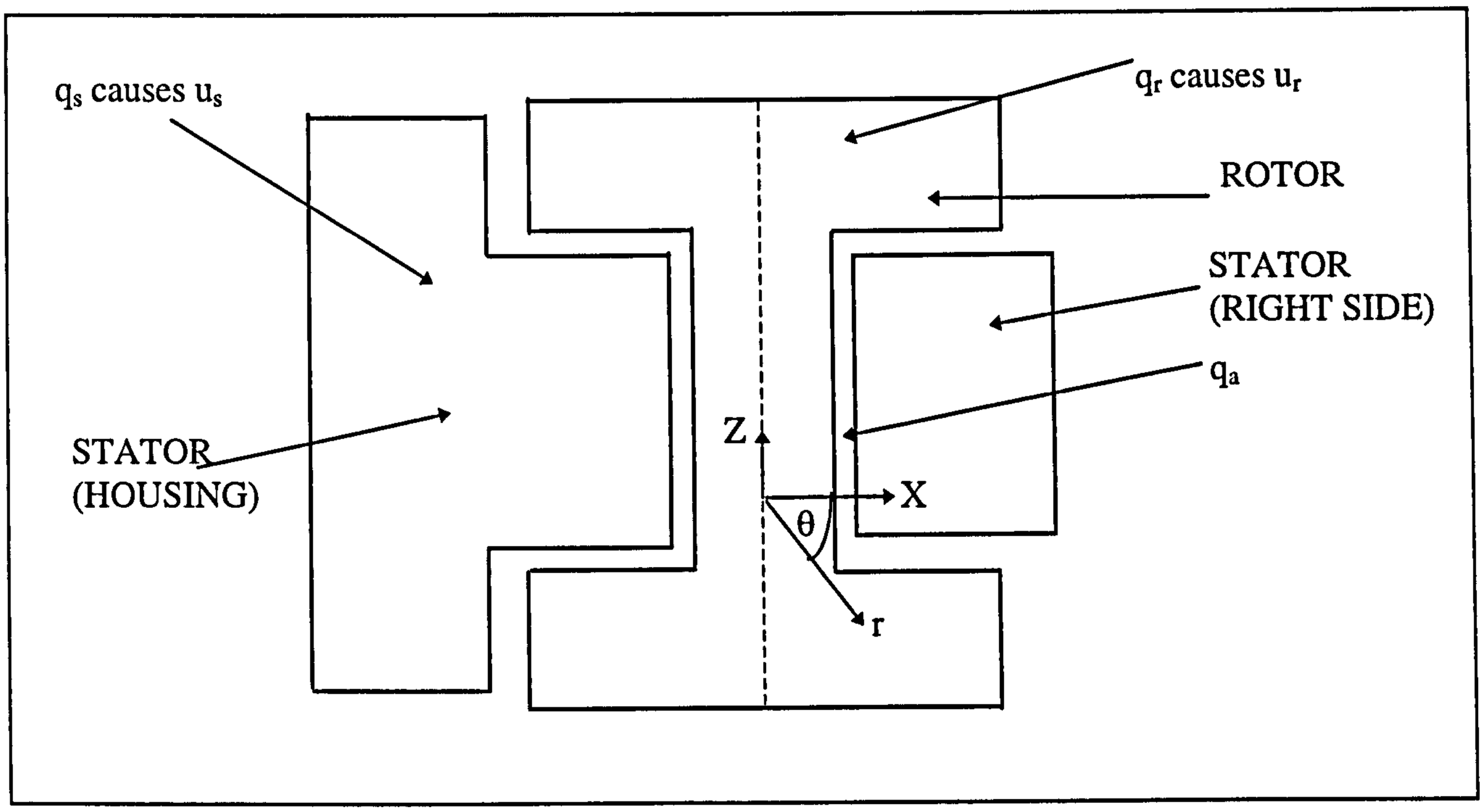


Figure 3.14: Thermal modelling of spindle

Figure 3.14 shows a simple model of the heat generation / transfer for the spindle used in this work. As a result of heat quantities (q_r) and (q_s), the bearing rotor and stator will exhibit *thermal growths* (u_r) and (u_s) respectively in the axial (Z) and radial (r) directions (as shown in figure 3.14). These thermal growths will influence journal and thrust clearances of the bearing, and subsequently affect bearing pressure distribution resulting in spindle error motions.

3.6.4.4 REDUCTION OF THERMAL EFFECTS

Thermal effects in spindles can cause spindle thermal growths of a few nanometers to 100 μm or more (Mckeown et al, 1995). In very high precision machine tools and measuring instruments it is important to reduce thermal effects to a very low level. This can be achieved by:

- knowledge of the influence of different heat sources within the machine structural loop by extensive thermal modelling eg using finite element methods or by empirical methods (Mckeown et al, 1995).

- use of ceramic (eg reaction bonded silicon nitride) or composites (eg carbon fibre reinforced plastics) spindles reduces thermal errors quite significantly as demonstrated by Hoffmann et al (1988).

- internal and external temperature control of the machine environment using oil or gas cooling and adequate air-conditioning, respectively. Mckeown et al (1995) have shown that the axial thermal growth of a spindle can be controlled to better than 20 nm within minutes of start up by using oil cooling.

- direct thermal error compensation (Hoffmann et al, 1988); which involves measuring the thermal drift in the machine or instrument and correcting the error in real-time.

CHAPTER 4: POTENTIAL TECHNIQUES FOR MEASURING SPINDLE RUN-OUT

4.1 INTRODUCTION

There are a number of techniques used for measuring error motions of spindles. Selection of a particular technique depends on sensor requirements such as linearity, dynamic range, resolution, sensitivity, frequency response and costs. Typical techniques for measuring spindle run-out are:

- Inductive sensing using an LVDT (Linear Variable Differential Transformer) probe (Slocum, 1992 pp174 - 176) , an eddy current probe (Orcutt, 1971)
- Capacitive sensing (Chapman 1985, Hansen 1988 , Yao-Sun 1985).
- Optical diffraction sensing (Gee et al, 1988).
- Fibre optic displacement sensing (Mizuno, 1993).
- Moiré technique (Kim & Park, 1993)
- Interferometric sensing (Gee et al 1988 , Triandis et al 1991, Idowu & Gee 1996).

An early part of this chapter (sections 4.2 to 4.5) will consider the non-interferometric techniques mentioned above, while the latter part (sections 4.6 to 4.7) will concentrate on different types of interferometric techniques for measuring spindle run-out, giving a detailed account of the Fizeau interferometer used in this project. In the final part of the chapter (section 4.8) a comparison of the different techniques for measuring spindle error motions is given.

4.2 CAPACITIVE SENSING OF SPINDLE RUN-OUT

4.2.1 INTRODUCTION

Capacitive gauging techniques are used for sensing spindle error motions due to their simplicity, non-invasive nature, high resolution and good frequency response characteristics.

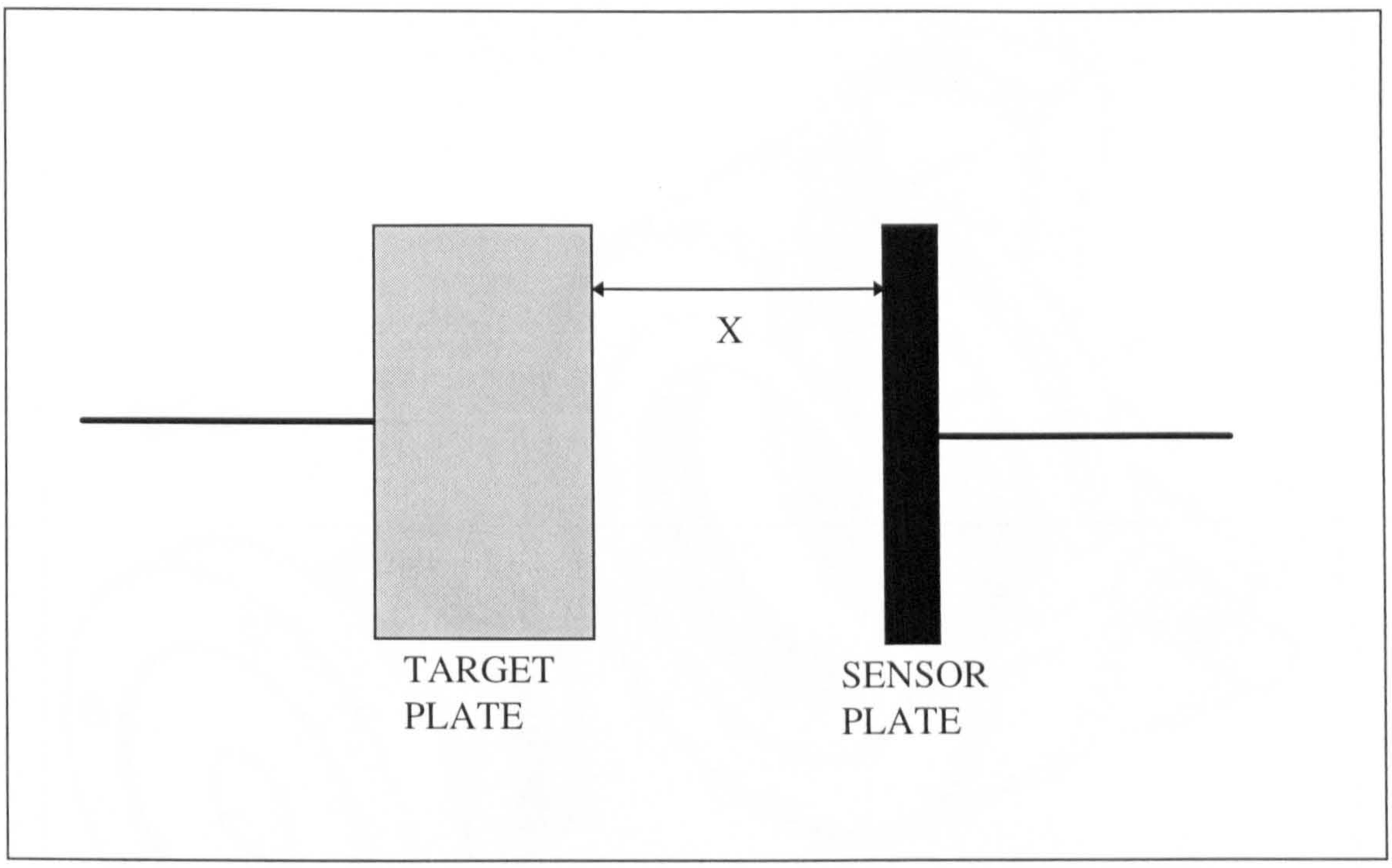


Figure 4.1: Parallel plate configuration of a capacitive sensor

4.2.2 THEORY OF CAPACITIVE SENSING

Most commercial capacitive sensors use the parallel-plate capacitor configuration which incorporates a variable air gap, as depicted schematically in figure 4.1. The capacitance of a parallel-plate capacitive sensor is:

$$C = \frac{\varepsilon_o \varepsilon_r S}{X} = \frac{8.85 \times 10^{-12} S}{X} \tag{4.1}$$

where, $\varepsilon_o \cong 8.85 \times 10^{-12}$: permittivity of free-space (Fm^{-1})
$\varepsilon_r = 1$: permitivity of air gap
S: effective area of plates (m^2)
X: distance between capacitance plates (m)

Sensitivity of the capacitance sensor is obtained by differentiating equation (4.1):

$$\frac{dC}{dX} = \frac{8.85 \times 10^{-12} S}{X^2} \quad (Fm^{-1}) \tag{4.2}$$

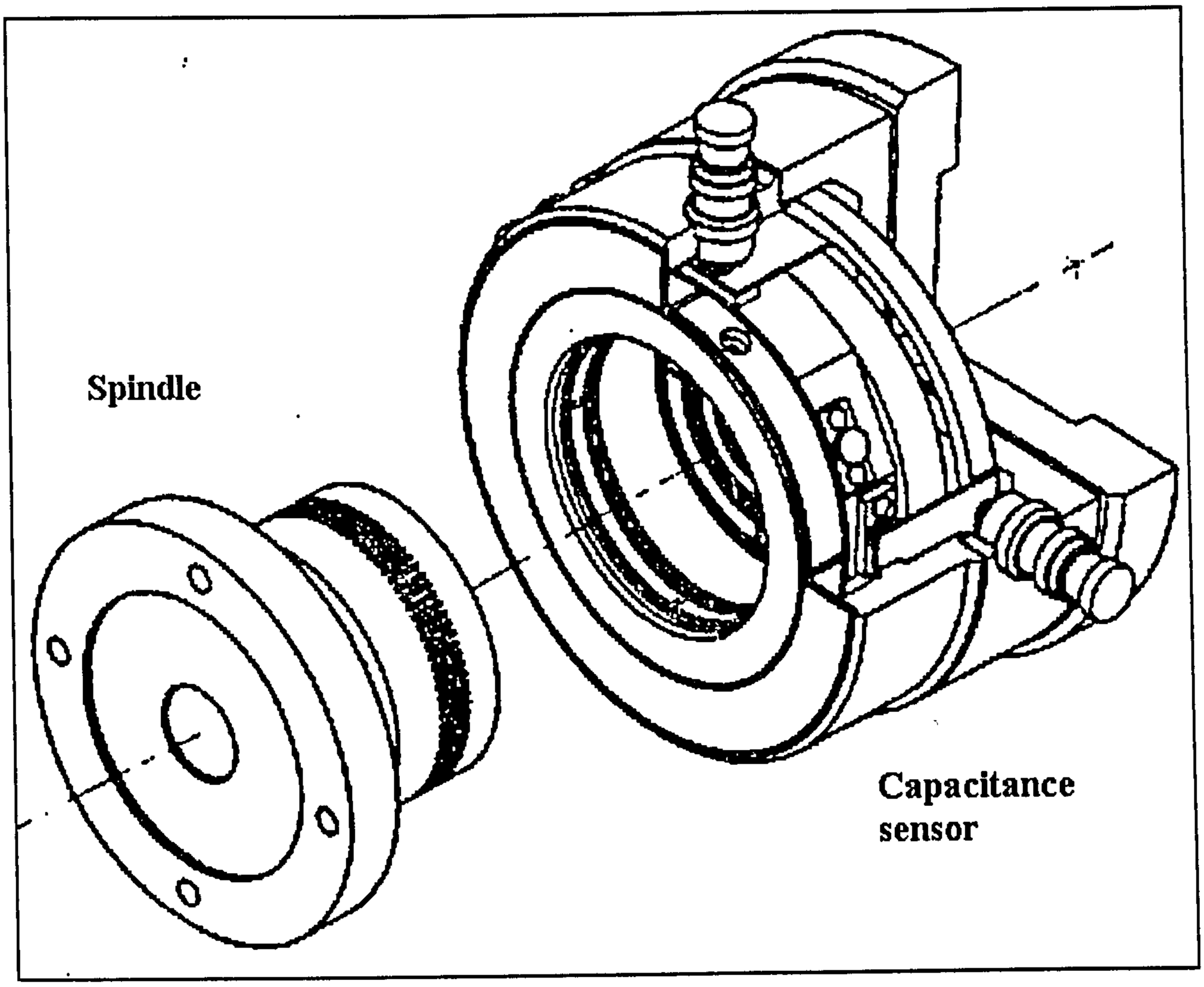


Figure 4.2 : Configuration of the capacitive spindle analyser after Chapman (1986)

The relationship between the voltage (V) and the charges (q) is:

$$V = \frac{q}{C} = \frac{It}{C} \qquad (4.3)$$

and, by substituting equation (4.3) into (4.1):

$$V = \frac{ItX}{8.85 \times 10^{-12} S} \qquad (4.4)$$

where I = capacitor charging current (A)
t = charging time (s)

If I, t and A are kept constant, the output voltage (V) would be linearly proportional to the plate separation X (Chapman, 1986).

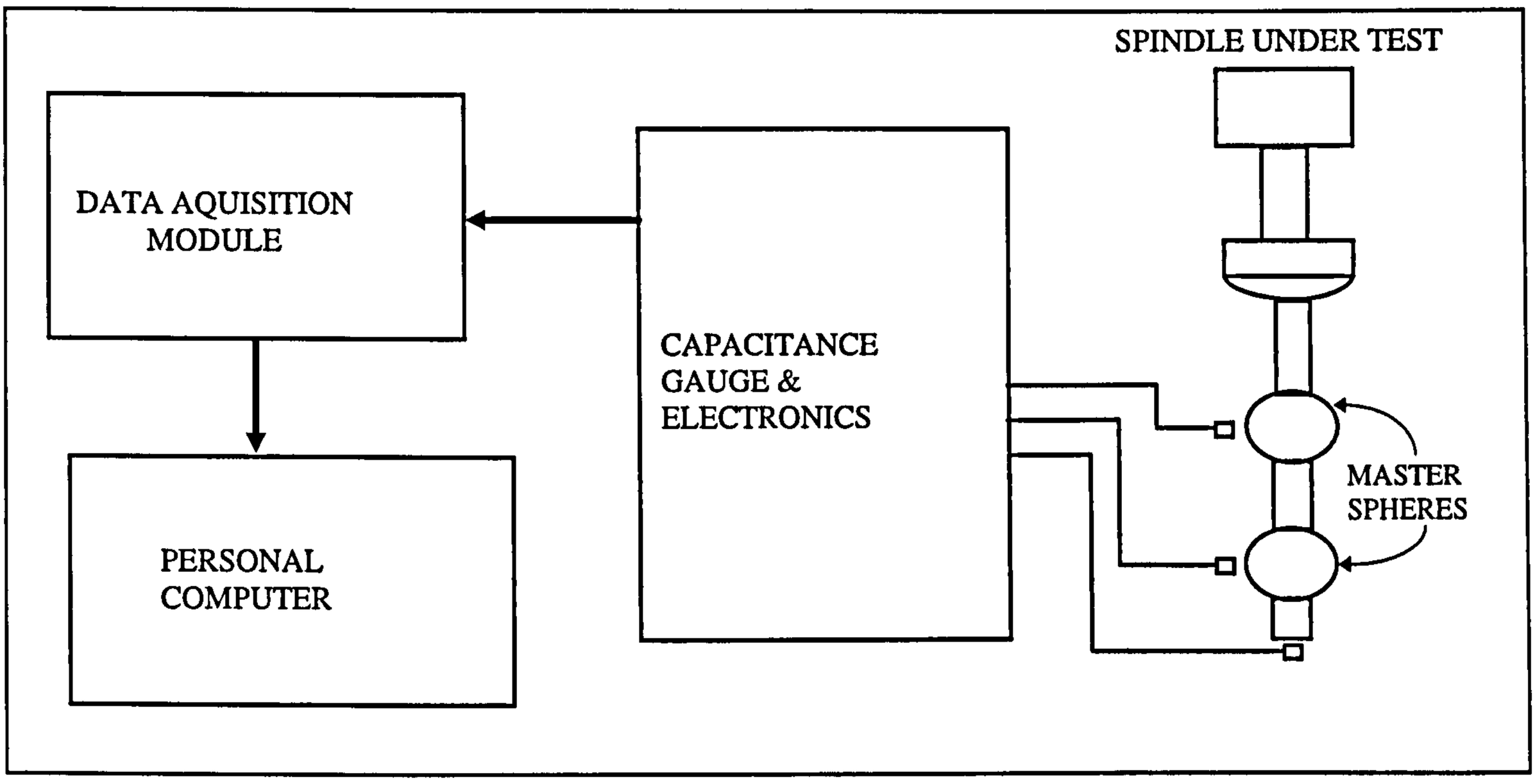


Figure 4.3 : Capacitive spindle analyser developed by Hansen (1988)

4.2.3 EXAMPLES OF CAPACITIVE SPINDLE ANALYSERS

A capacitive spindle analysis arrangement was developed by Chapman (1986) to assess the performance of a new experimental ultra-precision air-bearing spindle built by Cranfield Unit for Precision Engineering (CUPE)*. The measurement system incorporates a precision cylindrical master artefact that is carefully positioned coaxially onto the surface of the spindle undergoing the analysis. The capacitive sensor assembly is fitted onto the circumference of the master cylinder (as shown in figure 4.2), to maintain a gap of at least 50 μm. It incorporates eight identical capacitors, arranged as two quadrants. Each quadrant is used for measuring the respective radial and tilt error motions of the spindle. An additional capacitive sensor was also used to measure the spindle axial error motion. The capacitance probe (array) is initially charged-up with a specified current for a finite amount of time, and the voltage across each individual sensor is sampled and their output is a linear function of the spindle run-out in its respective degree of freedom (ie air-gap between the capacitor sensors and the surface of the master artefact used).

Hansen (1988) developed a capacitive spindle analyser at the Lawrence Livermore National Laboratory**. This system was capable of detecting spindle error motions of about 25 nm operating between 20 - 2000 RPM. Run-out was detected by measuring the motion of two precision gauge balls mounted on a special fixture which allowed each ball to be positioned nominally concentric with the axis of rotation of the

---

* Cranfield Unit for Precision Enginerring Ltd (CUPE), Cranfield, Bedford MK43 OAL

** Lawrence Livermore National Laboratory , Livermore ,Callifornia U.S.A. 94550

spindle, as shown in figure 3.3. Three capacitive sensors where used in this set up, one to measure the axial error and the other two to measure the respective radial and tilt error motions .

4.2.4 SIGNAL PROCESSING REQUIREMENTS

The signal from a capacitive sensor is usually converted from capacitance to voltage. This can be achieved by using an AC bridge circuit, which is very popular for its very high precision measurements at low bandwidths (Smith & Chetwynd, 1992). Chapman (1986) developed an alternative voltage-to-capacitance conversion system based on the voltage-current relationship given in equation (4.4). This approach involved a periodic charging/discharging of the sensor, giving a DC output voltage that was sampled at 12 kHz.

Amplifiers, multiplexers, summing circuits, filters, and A-to-D converters are used for the analogue signal processing requirements of the sensor. A computer or dedicated micro-controller is used to collect and process data, and to develop any control algorithms required for the device (eg artefact on-line centring). Speed and memory capacity requirements must also be considered when selecting the data acquisition modules.

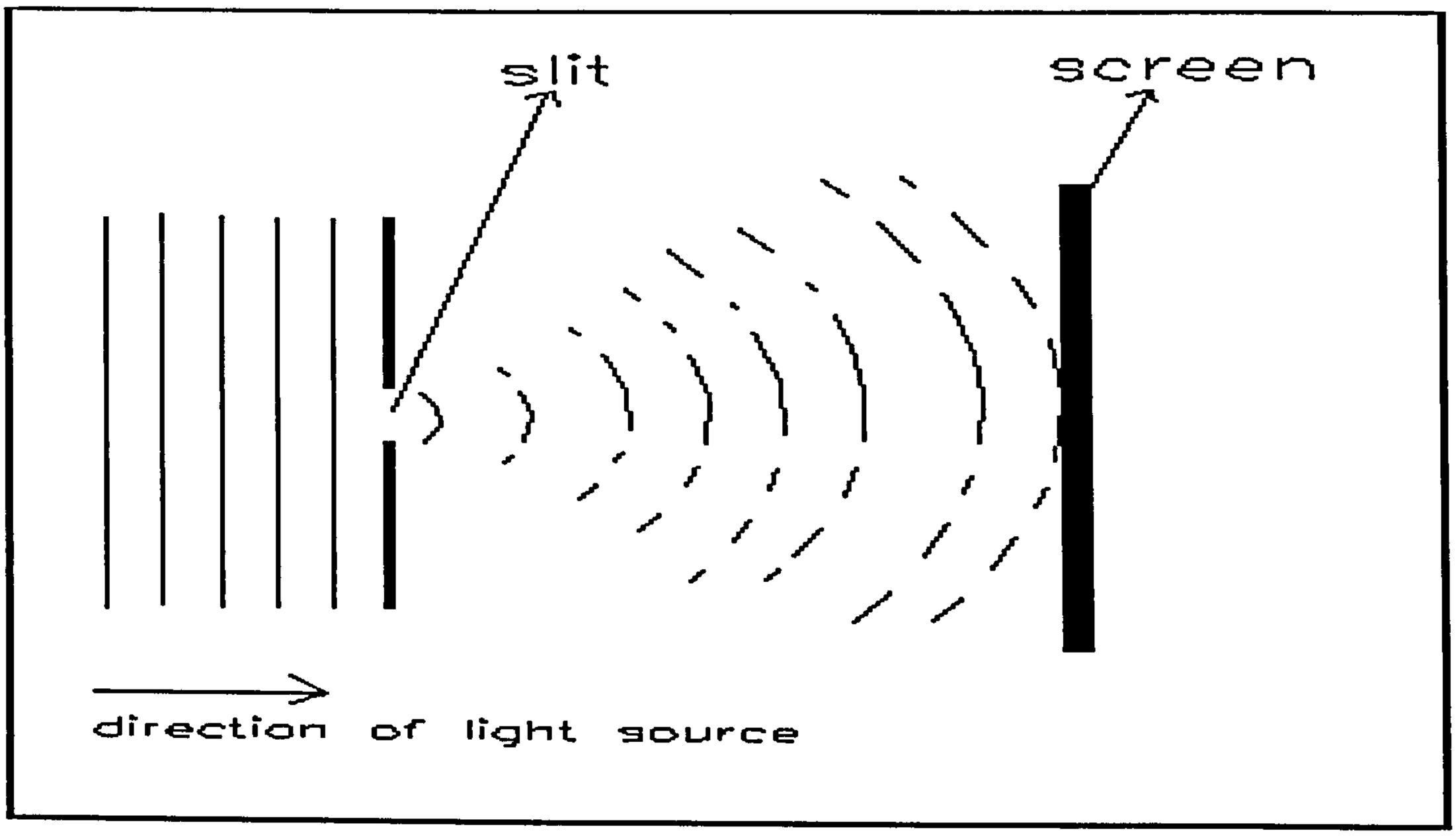


Figure 4.4 : The phenomenon of Fraunhofer diffraction

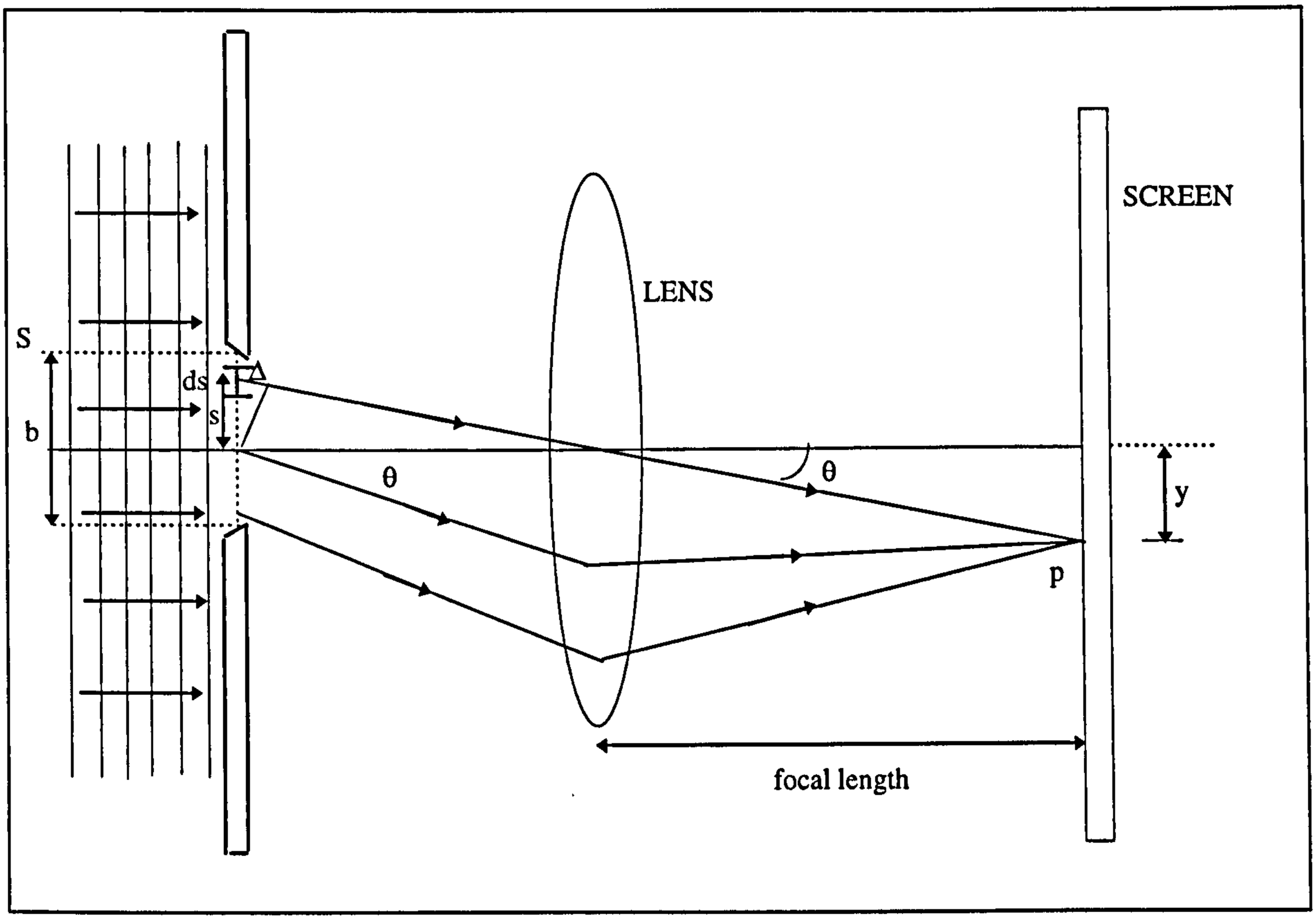


Figure 4.5 : Fraunhofer diffraction using Huygen's wave front construction

4.3 OPTICAL PROXIMITY SENSING OF SPINDLE RUN-OUT

4.3.1 INTRODUCTION

The optical proximity (diffraction) sensor described in this work incorporates the phenomenon of Fraunhofer diffraction. According to Pedrotti & Pedrotti (1993, pp324 - 330), Fraunhofer diffraction occurs when a source of light and an observation screen are effectively far enough from a diffraction slit, so that the wavefronts arriving at the aperture may be considered to be planar (as shown in figure 4.4). In this section a brief description of Franhaufer diffraction, design and performance of a optical diffraction proximity sensor are presented.

4.3.2 THEORY OF FRAUNHOFER DIFFRACTION BY A SINGLE SLIT

The classical approach to explaining the theory of Fraunhofer diffraction is by using Huygens wavefront construction. Figure 4.5 depicts a slit width (W) with finite elements (ds), and a light source S. Imagining the slit to consist of finite or strips of length (ds), it is possible to calculate the amplitude and intensity values of the diffraction pattern, by integrating the number of wavelets emitted by each of the finite strips. A make-up of all

the imaginary ds elements will contribute to a spherical wavelet at point (P) and of amplitude:

$$d E_p = \left(\frac{E_L\, ds}{r_o + \Delta}\right) e^{i[k(r_o+\Delta)-\omega t]} \quad \text{(Pedrotti \& Pedrotti, 1987 pp324 - 330)} \qquad (4.5)$$

where,

dE_p:	amplitude of the wavelet sources
E_L:	amplitude per unit length from source
r_o:	radial distance of the wavelets from slit
k:	propagation constant
w:	angular frequency
$\Delta = s \sin\theta$:	phase
t:	time

Total irradiance (I) at P is proportional to the square of the amplitude contributions of the wavelet sources emanating from the slit and, by integrating equation (4.5):

$$I = I_0 \frac{b \sin\beta}{\beta} = I_0\, b\,\mathrm{sinc}\,\beta \qquad (4.6)$$

where, $\beta = \frac{1}{2} bk \sin\theta$ (4.7)

represents the phase difference between the wavelets from the centre and either endpoints of the slit when s = b/2 (Pedrotti & Pedrotti, 1993 pp324-330)

$I = I_0$ is a maximum value as β approaches zero ie sincβ approaches 1. However, the intensity value I is zero at $\beta = n\pi$ (ie where n is an integer) these values represent the minimum points of the diffraction pattern. This graph shows a central bright band (ie the *central maximum*) accompanied by a series of narrow bands known as *subsidiary maxima*. Except for the central maximum lobe, The subsidiary maxima can be found by differentiating the intensity value I with respect to β:

$$\frac{d}{d\beta}\left(\frac{\sin\beta}{\beta}\right) = \frac{\beta\cos\beta - \sin\beta}{\beta^2} = \tan\beta = \beta \qquad (4.8)$$

The ratio of the central maximum lobe to the first minimum lobe is:

$$\frac{I_{\beta=0}}{I_{\beta=0.5\pi}} = \frac{\left(\sin^2\beta/\beta^2\right)_{\beta=0}}{\left(\sin^2\beta/\beta^2\right)_{\beta=0.5\pi}} = \frac{1}{\left(\frac{\sin^2\beta}{\beta^2}\right)_{\beta=0.5\pi}} = 3.95248 \approx 4 \qquad (4.9)$$

Hence the intensity of the angular width (which spans the two minima located either side of the central maximum lobe ie $-0.5\pi < \beta < 0.5\pi$) is approximately half the central lobe maximum intensity: $0.5I_o$. This value is proportional to a threshold voltage V', used to detect the occurrence of the angular width on a CCD photodetector.

4.3.3 DEVELOPMENT OF SENSOR USING *VIRTUAL* SLIT

Gee et al (1988) developed an optical diffraction sensor which incorporated a He-Ne (Helium - Neon) laser source that uses a fibre optic head to focus the beam on to a *virtual* slit (ie formed by the distance between a blade edge and a plane mirror surface, as shown in figure 4.6). This work demonstrated that the Fraunhofer diffraction patterns obtained from this arrangement can be readily quantified and processed. With the aid of a charged couple device detector (CCD) the *central maximum width* of the diffraction pattern intensity distribution can be detected and used to measure tool proximity. The same technique can also be applied to measure spindle run-out.

Figure 4.7 shows an optical diffraction arrangement used for sensing spindle run-out. The initial set-up of the slit width can done precisely by servoing a knife edge towards a plane mirror surface attached to the spindle, with the aid of a computer controlled stepper motor and lead-screw mechanism as shown in figure 4.7. When the beam from the laser is normal to the slit plane, $\beta = \frac{1}{2} bk \sin\theta$, and substituting $k = \frac{2\pi}{\lambda}$:

$$\beta = \frac{\pi b \sin\theta}{\lambda} \qquad (4.10)$$

where, b: width of the slit
θ: angle of diffraction
λ : Wavelenght of the light source

By modifying equation 4.10, the *half angular width* between the maximum of the central bright band and the nth dark band zero intensity is:

$$\sin\theta_n = \frac{n\lambda}{b} \qquad (4.11)$$

Equation 4.11 indicates an inverse linear relationship between the angular width and the slit width.

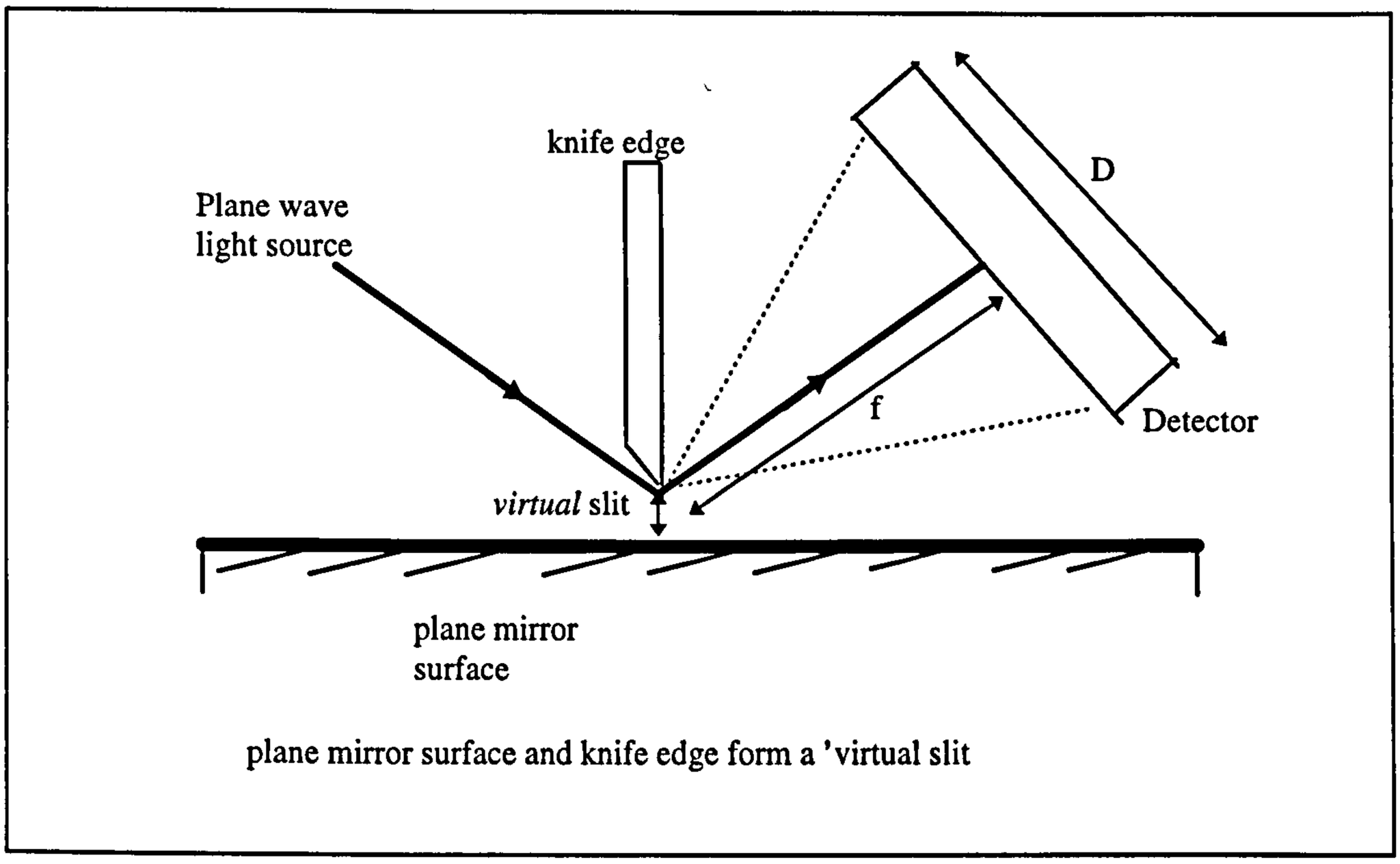


Figure 4.6: The *virtual* slit

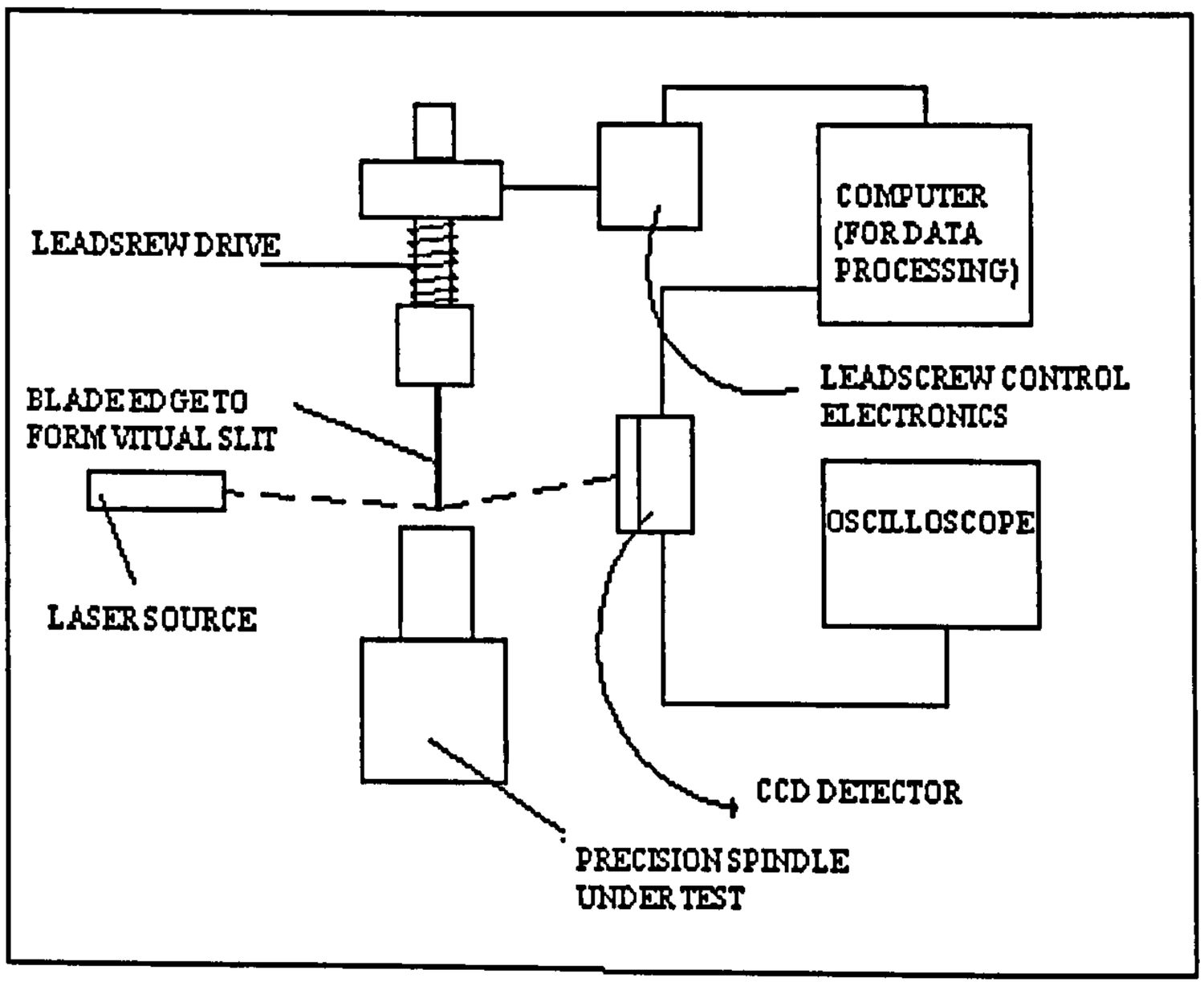


Figure 4.7 : Optical diffraction sensor set-up for spindle analysis

In order to sense the optical diffraction pattern, a 256 - photosite linear CCD array was used by Gee et al (1988). Typical CCD line-scan photodetector arrays of 256 - sites, 512, 1048, 1728, 2048 or 4096 sites in single-chip IC form are commercially available from suppliers such as Dalsa Inc*. These devices typically have scanning rates of 10 to 80 MHz and photoelement size of 14μm × 14μm for moderate speed photo-detectors, and 7μm × 7μm for the very high speed photo-detectors. A line scan photo-detector of 512 photosites used at 50 MHz for instance, should be applicable of achieving a sampling (line) rate approximately of 100 kHz, which is adequate for most spindle metrology applications.

4.3.4 SIGNAL PROCESSING CONSIDERATIONS

The Franuhofer diffraction signal is sampled at specified intervals corresponding to the photosite pitch of the CCD photodetector array. Measurement of the number of photosites contributing to the angular width (ie corresponding to the spindle sub-micrometre run-out) is done when the output voltage of the CCD is greater than or equal to the threshold voltage V' (see section 4.3.2). A method for calculating the number of photosites corresponding to the angular width was devised by Green (1987), in which the number of elapsed periods of the CCD clock signals was measured while the photodector output signal was above V'.

Figure 4.8 shows a simplified schematic of the signal processing components used by (Green, 1987) for measuring the angular width. According to Pedrotti & Pedrotti (1993, pp324-330) the CCD width (D) required to accommodate the central maximum lobe and the first minima is

$$D = f\sin\theta = \frac{f\lambda}{b} \qquad (4.12)$$

where, f: distance between silt and screen (focal length)

By modifying equation (4.12), the slit width (formed by the knife edge and reflective surface) is:

$$b = \frac{xf\lambda}{D} = \frac{xf\lambda}{Np} \qquad (4.12)$$

where, D: width of CCD to accommodate maximum lobe and 1st minima

λ: wavelength of laser source
N: number of photosites detecting angular width (ie number counts)
p: width of CCD photosite

Sensor uses the linear relation between b and N.

---

* Dalsa Inc. 605 McMurray Road, Waterloo, Ontario, Canada N2V 2E9

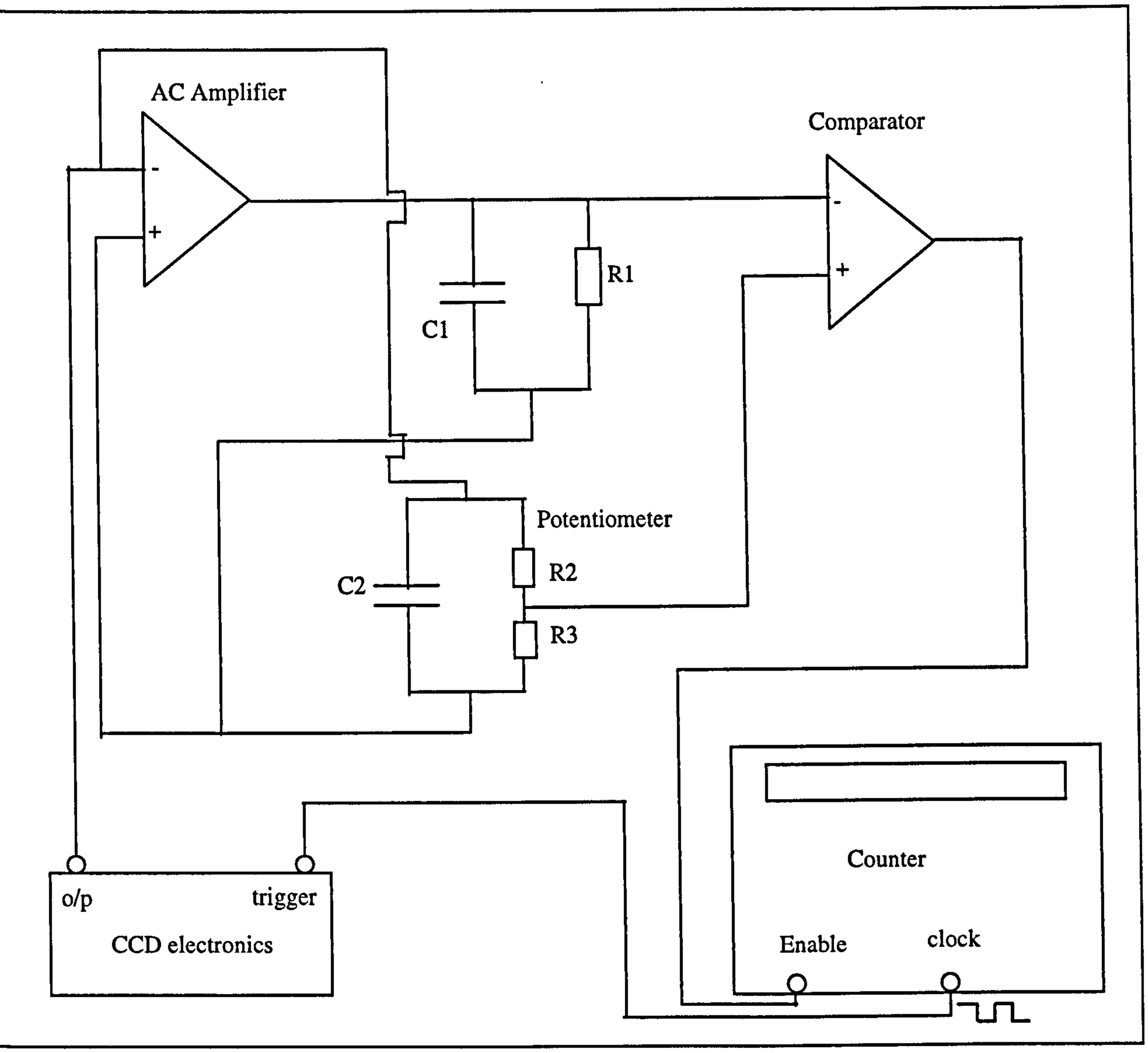


Figure 4.8: Signal processing for CCD diffraction sensor (after Green, 1987)

4.4 FIBRE OPTIC SENSING OF SPINDLE RUN-OUT

4.4.1 INTRODUCTION

Fibre optic sensing is increasingly replacing some of the conventional methods of sensing. This is due to the benefits gained by using fibre optic sensors (Slocum, 1992 pp268 - 269), which include the following:

- insensitive to electromagnetic interference due to the lack of electrical bias at the sensor
- capability of operating in very harsh environments
- very good frequency response over a wide range
- very high accuracy
- readily interfaced with semiconductor devices
- operation at low power regimes

In this section a brief description of the theory, design and signal conditioning requirements of backscatter fibre optic sensors are given.

4.4.2 THEORY OF FIBRE OPTIC (BACKSCATTER) SENSING

According to Doebelin (1990, pp268 - 269), a fibre optic backscatter sensor consists of a bundle of several hundred optical fibres, each a few micrometres in diameter. The fibre bundle is divided in to two equal groups of fibre. One group known as the *transmitting fibre*, carries the light to the probe tip, where it is scattered and reflected by a target surface. Some of the reflected light is picked up by the other group of fibres known as the *receiving fibre*, and is transmitted and focused on a suitable photodetector whose electronics then produces a dc output related to probe-target proximity.

There are generally three types of fibre optic configurations, these include:

- Hemispherical
- Random
- Fibre pair

Figure 4.9 shows a schematic of these fibre configurations and, a graph of the reflection against distance (Slocum, 1992 pp174 - 176) indicating the *front* and *back-slope* characteristics of the sensor. The front-slope rises at very close proximity to the target surface, where the output rises with distance.. The Back-slope on the other hand indicates a decrease in intensity as the distance between the sensor and the target surface increases.

Figure 4.10 shows a schematic of a backscatter fibre optic sensor used for measuring spindle run-out. An example of the use of fibre optic sensors for precision measurement, is the arrangement of Mizuno (1993) in which three fibre optic probes were used to measure error motions (the two orthogonal tilts and axial error motions) of an ultra precision spindle. The fibre optic probes used in this work had bent tips, to improve accessibility.

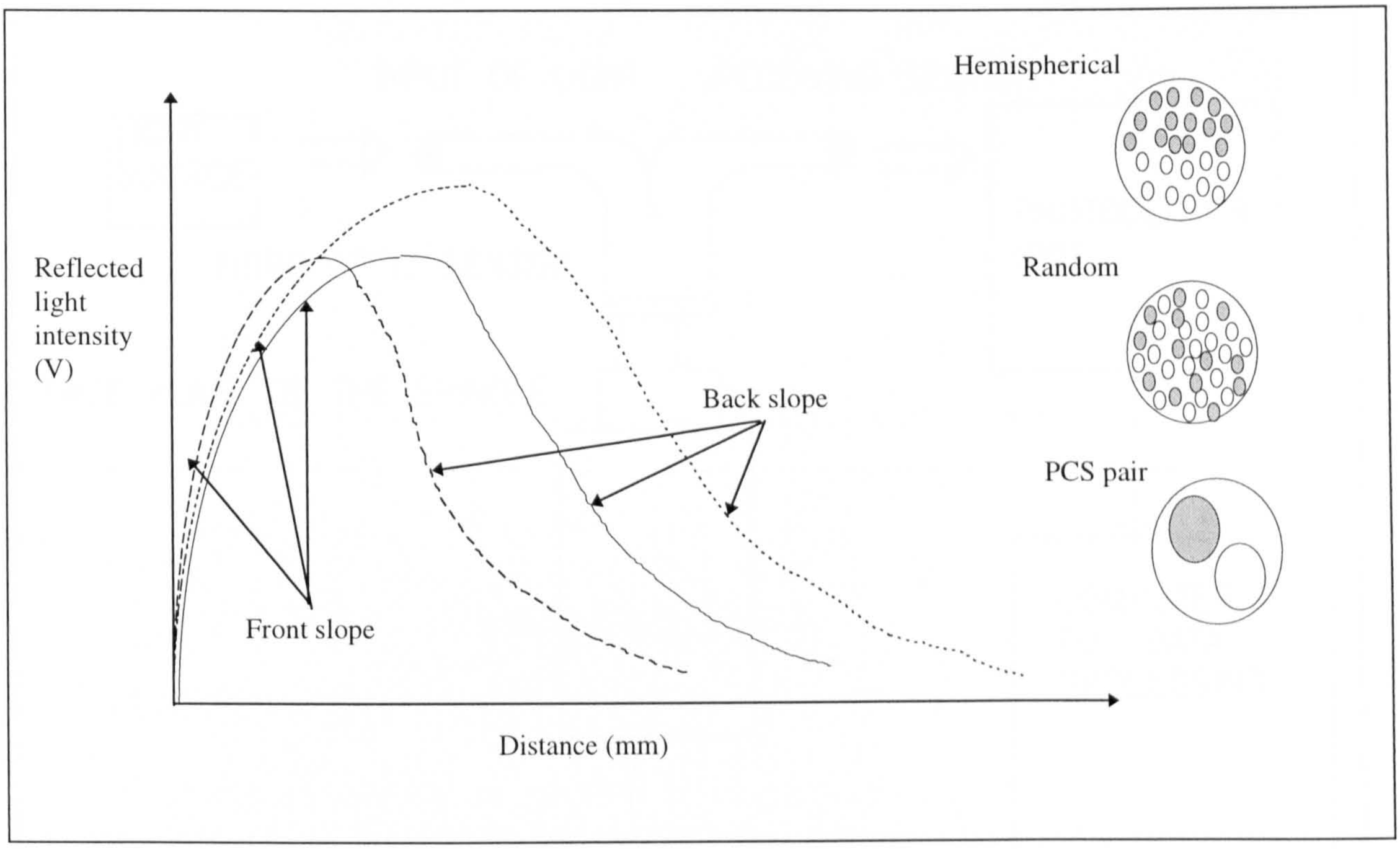


Figure 4.9: Fibre configurations and characteristics, courtesy of Slocum (1992)

4.4.3 SIGNAL PROCESSING REQUIREMENTS

According to Culshaw & Dakin (1989) the signal obtained from a fibre optic sensor is modulated in response to the measurand field (ie displacement in this case). The (received) electric field signal can be represented by the following relationship:

$$\mathbf{E_r}(\lambda) = \mathbf{T}(\mathbf{X}, \lambda)\, \mathbf{E_s}(\lambda) \qquad (4.13)$$

where, $\mathbf{E_r}$: Electric field matrix at the receiver unit
$\mathbf{E_s}$: Electric field at the source unit

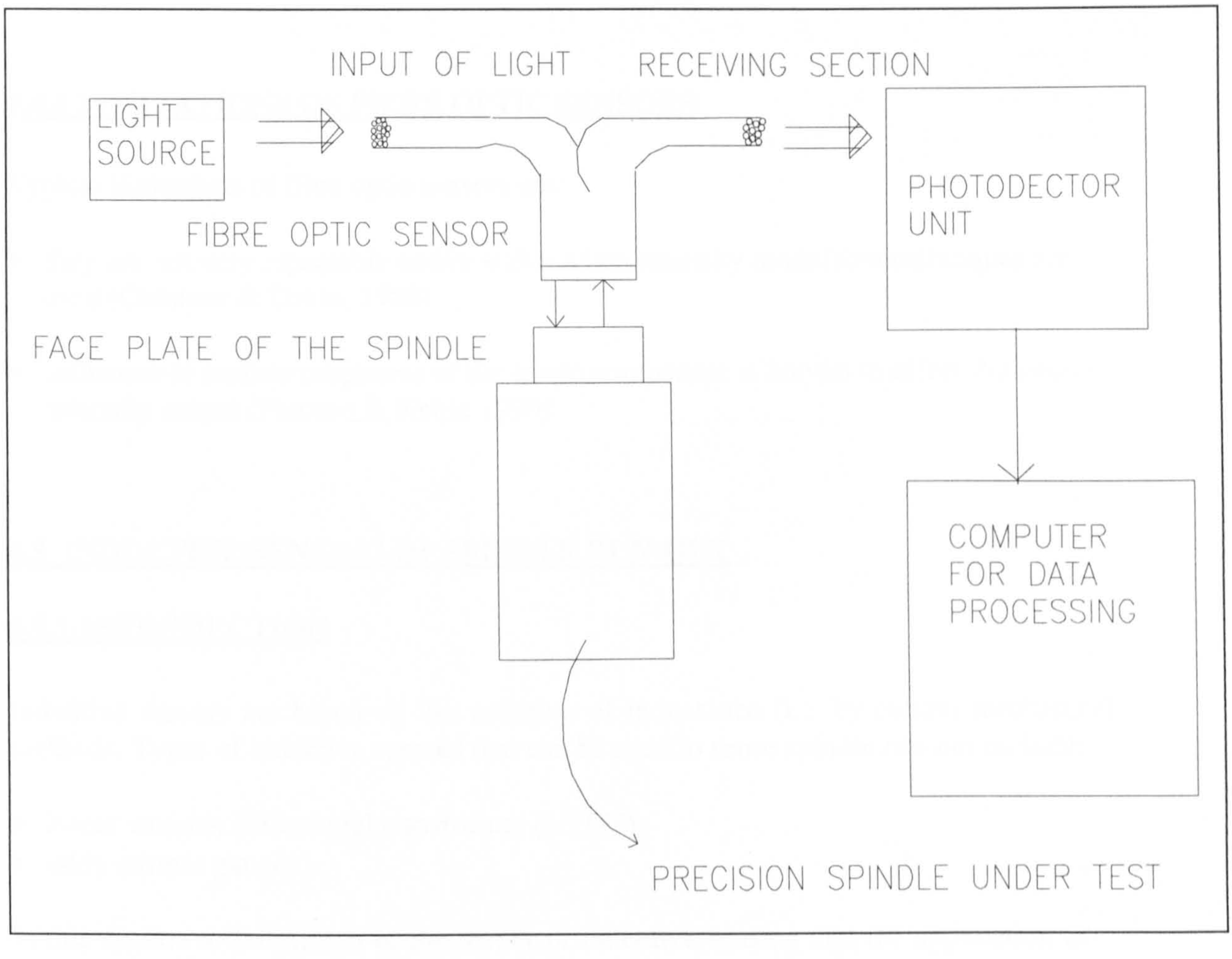


Figure 4.10: Configuration of the fibre optic sensor after Mizuno (1993)

X: Vector describing the physical environment of the measurand
λ: Wavelength of the source light
T: Sensing element (ie the fibre optic sensor) propagation vector

In the case of measuring spindle run-out, an intensity-modulated fibre optic sensor can be used (as shown in figure 4.10). The modulation mechanism for this sensor is the mechanical displacement in required degrees of freedom of the spindle reference plate see Culshaw & Dakin (1988) for more details on intensity modulators.

Typical signal processing components for an intensity modulated fibre optic sensor would include :

- signal amplifiers
- low pass and high pass filters
- attenuators
- display unit
- microprocessor unit

4.4.4 LIMITATIONS OF FIBRE OPTIC SENSORS

Typical limitations of fibre optic sensors are:

- they are not very repeatable and/or stable when intensity modulated techniques are used (Culshaw & Dakin, 1988)

- influence of surface roughness of the tested component is known to affect the sensors intensity output (Fawcett & Keltie 1990)

4.5 INDUCTIVE SENSING OF SPINDLE RUN-OUT

4.5.1 INTRODUCTION

Inductive sensors are based on the variation of inductance (L), by certain mechanical methods. Types of inductive sensors that can be used to sense spindle run-out include:

- linear variable differential transformer (LVDT)
- eddy current gauges

In this section a description of the theory of inductive sensing and the application of LVDT and eddy current sensors in the measurement of spindle error motions is given.

4.5.2 THEORY OF INDUCTIVE SENSING

Bentley (1988) described an LVDT inductive sensor as a transformer with a single primary coil energised by an AC voltage (as shown in figure 4.11)

$$V_s = V\sin 2\pi f_s t \qquad (4.14)$$

where, V: amplitude of supply signal
f_s: frequency of supply signal
t: time

and two secondary coils whose values are V_1 and V_2 respectively.

The output signal of the sensor (V_{out}) is

$$V_1 - V_2 = V \sin(2\pi f_s t + \varphi) \qquad (4.15)$$

where, V_1: signal on the first secondary coil.

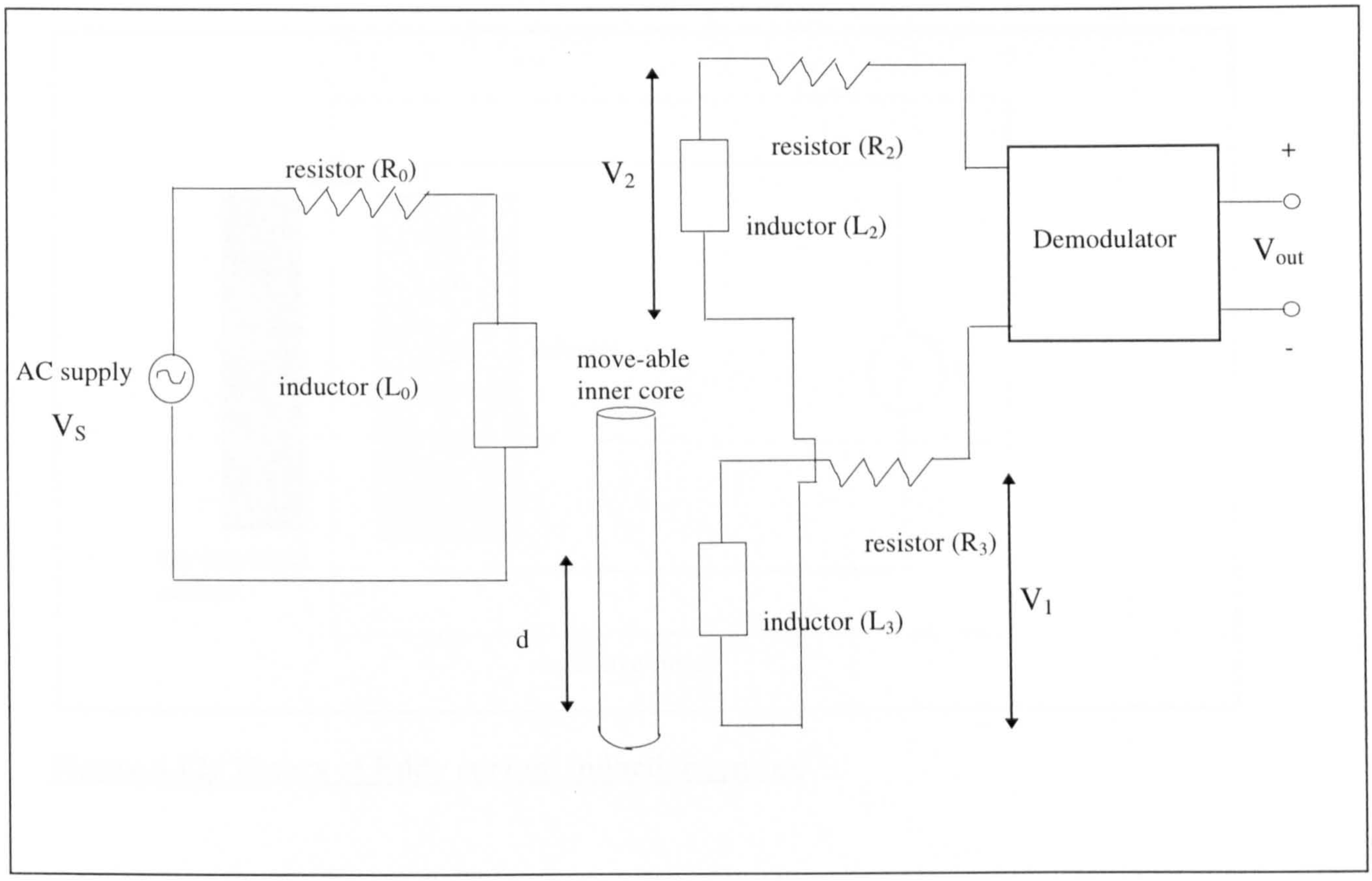


Figure 4.11: Principle of LVDT sensing

V_2: signal on the second secondary coil.
φ: phase change
V_o: amplitude of output signal

Operation of the eddy current probe on the other hand is based on Faraday's law of electromagnetic induction, where the induced electromagnetic force (EMF) in an eddy current sensor (which has a coil of (n) turns) is proportional to the rate change of magnetic flux (ϕ) generated in the sensor:

$$E \infty \frac{d}{dt}(n\phi) \qquad (4.16)$$

Figure 4.12 shows the principle of an eddy current sensor, the value E varies as the conducting target surface approaches the inductive circuit, and the linear characteristics of the displacement (d) of the target surface with respect to the sensor output (E) can be calibrated.

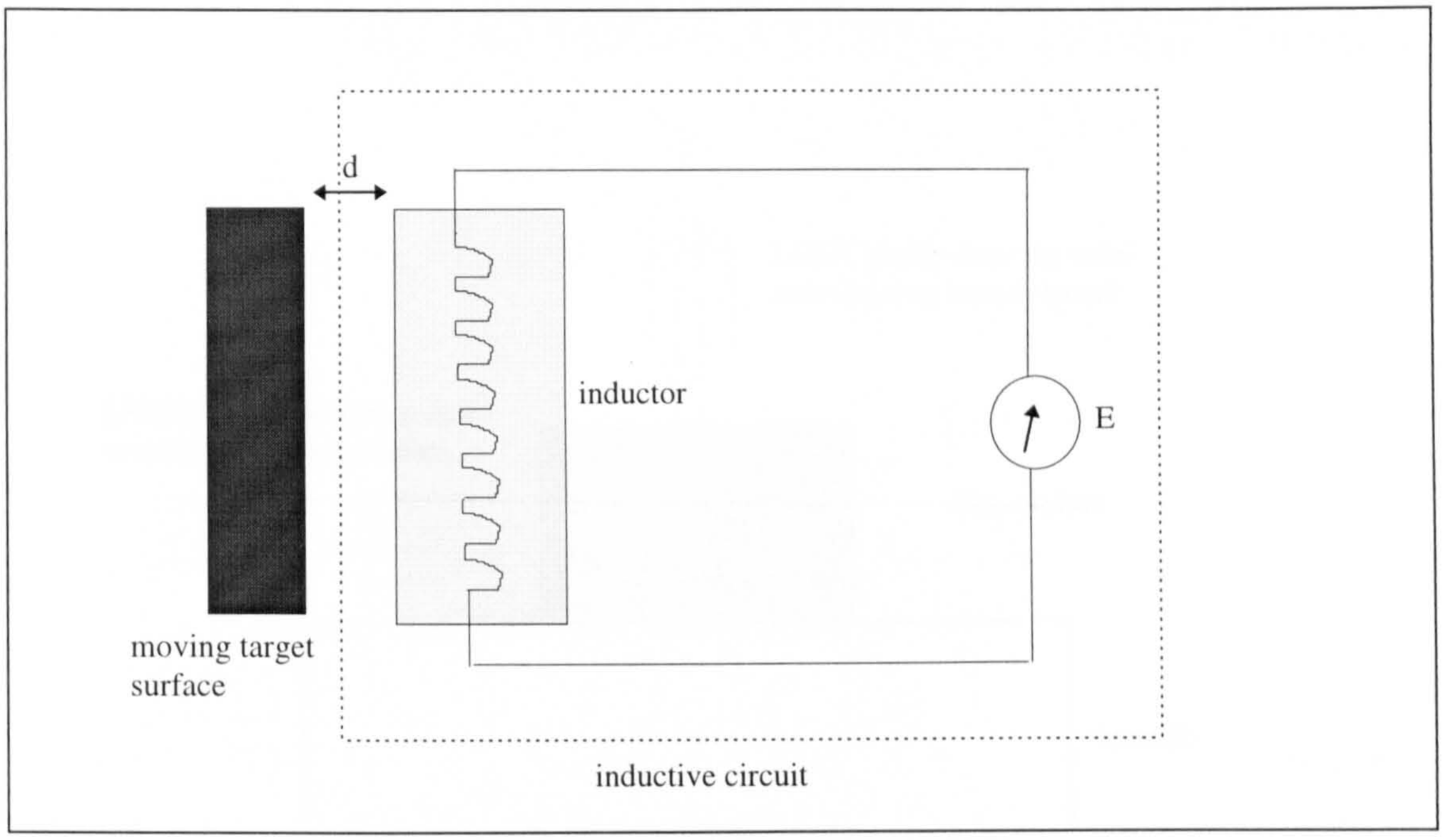


Figure 4.12: Theory of Eddy current inductive sensing

4.5.3 SIGNAL PROCESSING FOR LVDT (CONTACTING) SENSORS

An LVDT probe has one primary and two (identical) secondary coils, it also incorporates an inner core as shown in figure 4.11. The primary coil is excited with a sinusoidal voltage of 3V to 15V rms amplitude, and 60 Hz to 20,000 Hz frequency (Doebelin, 1990 pp233-242). This frequency induces a sinusoidal signal on two secondary coils, and the amplitude of this signal (e_o) varies linearly with the position (d) of the inner core. The output from the sensor is a modulated signal which is demodulated and filtered. The LVDT sensor is a contact probe (see figure 4.13), requiring mechanical contact between the measured object (master artefact attached to spindle) and probe tip coupled to its inner core. An application of a LVDT probe for measuring spindle error motions is given by (Slocum, 1992, pp543 - 550), in which the probe was used to measure the run-out of a tapered roller bearing (with a minimum axial error motion of 23 μm and operating at a maximum spindle speed of 700 rpm). Due to their (mechanical) contact with the measured artefact, LVDT probes are usually not suitable for sub-mirometre spindle run-out measurements.

4.5.5 SIGNAL PROCESSING FOR EDDY-CURRENT (NON-CONTACTING) SENSORS

An eddy-current probe uses two coils namely *active* and *balance* (shown schematically in figure 4.14), the first coil is influenced by the presence of a conducting target surface (measured artefact) and the second coil is used to provide temperature compensation (when connected to an AC bridge circuit).

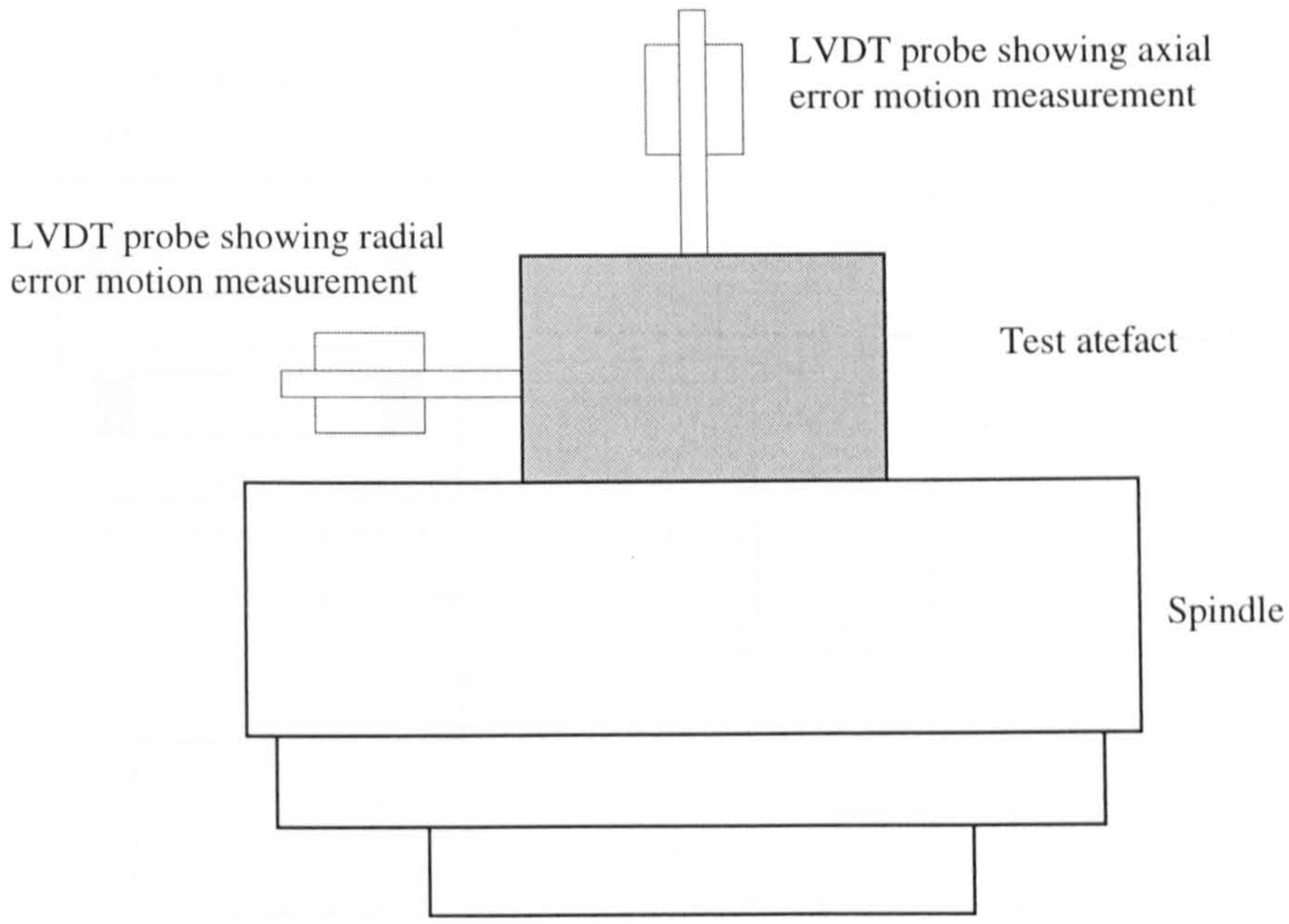


Figure 4.13: Schematic of LVDT probe

The probe AC bridge is usually excited using a 1 MHz sinusoidal signal (Doebelin, 1990, pp251 - 253). Magnetic flux lines from the active coil penetrate the measured artefact surface (which must be electrically-conducting), producing eddy currents whose density varies with respect to the proximity of the surface. Figure 4.15 depicts sensor output versus surface displacement characteristics of the instrument, acceptable linearity of the probe can be achieved by restricting the operating range to the mid-section of the output curve (Orcutt, 1971).

Output signals from the probe are usually demodulated and amplified using a log-amplifier (which provides more amplification as the signal gets weaker).

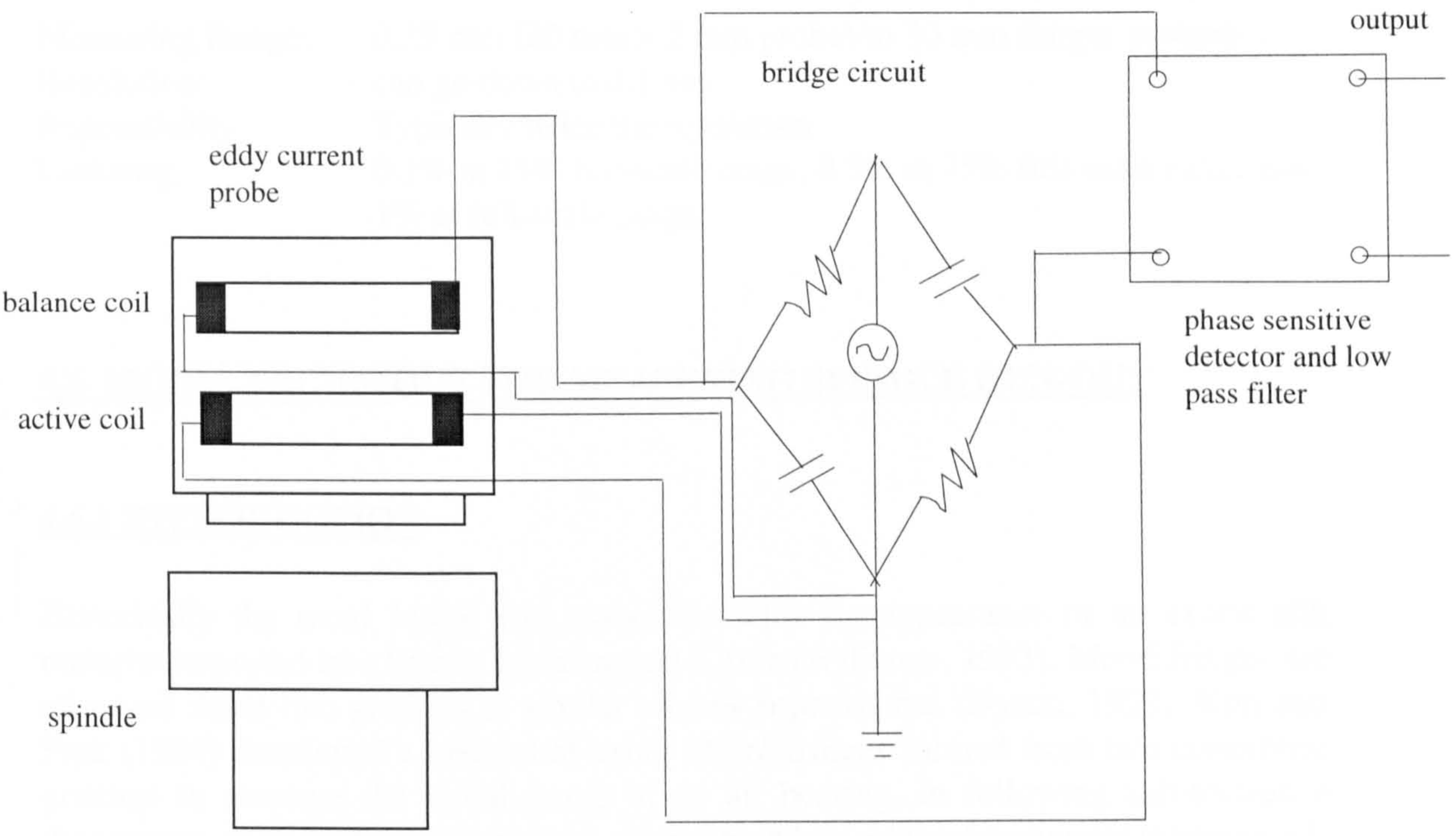


Figure 4.14: Schematic of Eddy-current sensor after Doebelin (1992 pp251 -253)

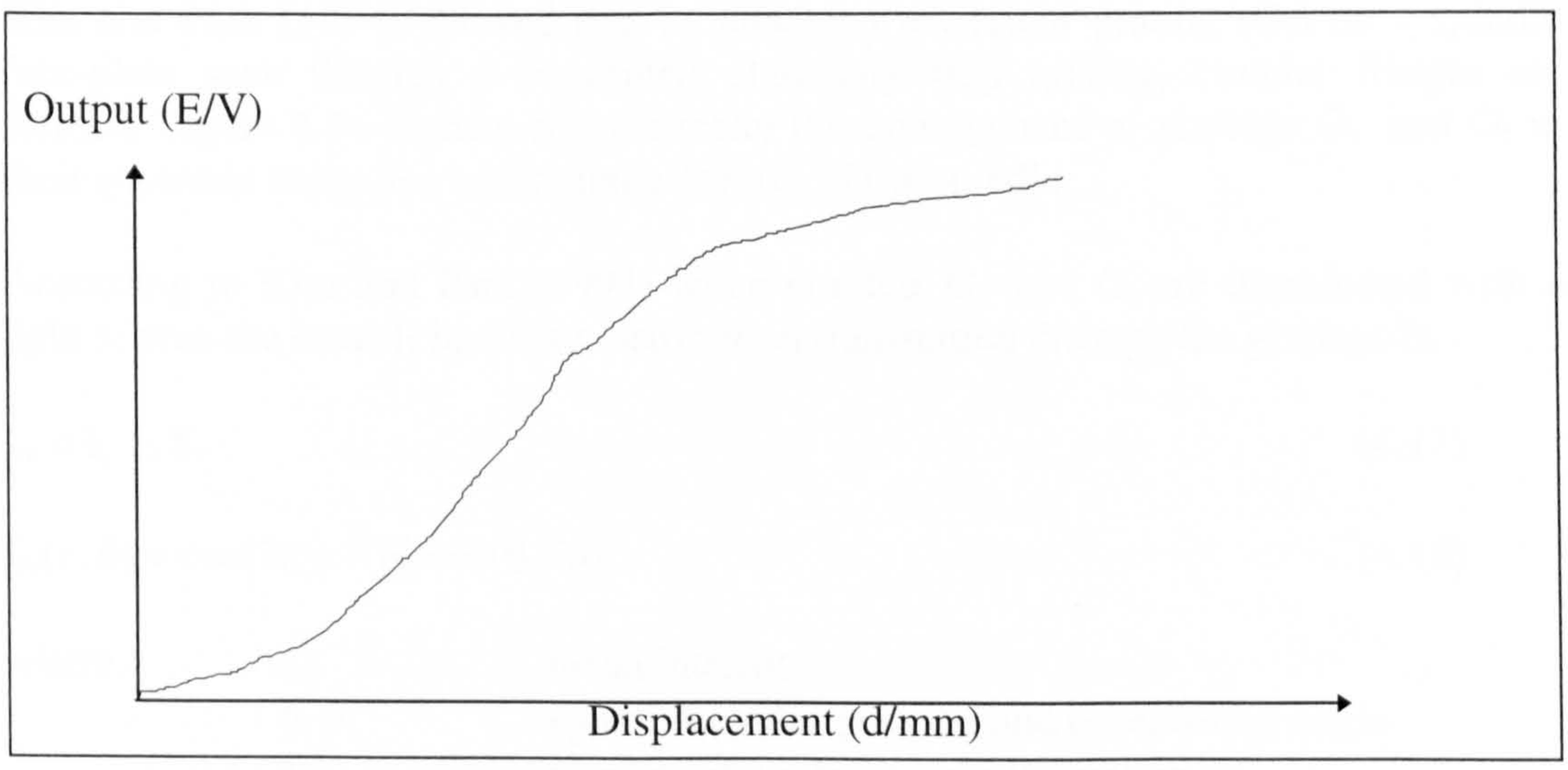


Figure 4.15: Output characteristics of Eddy-current sensor after Orcutt (1971)

These probes are commercially available from Kaman Instrument Corporation* with typical general characteristics:

Measuring Range:	0.25 mm (20 mm × 2 mm probe) to 30 mm (larger probes)
Resolution:	can go down to 0.1 nm
Repeatability:	Typically twice the resolution
Linearity:	0.1% at 25% full-scale range, 0.5% at 75% full-scale range and 1% at full-scale range.

4.6 MOIRÉ TECHNIQUE FOR MEASURING SPINDLE RUN-OUT

4.6.1 INTRODUCTION

Historically the word Moiré was associated with the appearance of an exotic silk material imported into France from ancient China (Williams, 1993). Moiré fringes are observed when two gratings of similar pitches superimpose (Dyson, 1958). Kim and Park (1994) developed a method of using Moiré fringes formed from two concentric gratings to measure the radial errors of an air bearing. In following sub-section a description of a spindle measurement system based on a Moiré technique is presented.

4.6.2 THEORY OF MOIRÉ (CIRCULAR) FRINGE FORMATION

Kim and Park (1994) showed that by attaching a circular grating (G_s) on a spindle face-plate seen through a concentric stationary (G_r) grating, circular fringes are formed. Figure 4.16 depicts schematically the arrangement of gratings G_s and G_r in their eccentric state, due to a radial offset (e) of the spindle.

According to Kim and Park (1994) when gratings G_s and G_r are illuminated with a light source the overall intensity distribution transmitted through the gratings is:

$$I_m = I_o S_r S_s \qquad (4.17)$$

$$I_m(\mathbf{r}, \theta) = \cos(2\pi)\mathbf{r} - \mathrm{Necos}(\theta - \phi) \qquad (4.18)$$

where,	I_o:	mean intensity
	ϕ, θ:	spindle angular position and eccentricity angle
	N:	number of circles in spindle grating
	R:	radius of gratings G_r and G_s

* Kaman Instrumentation Corp., Colorado Springs, Co.

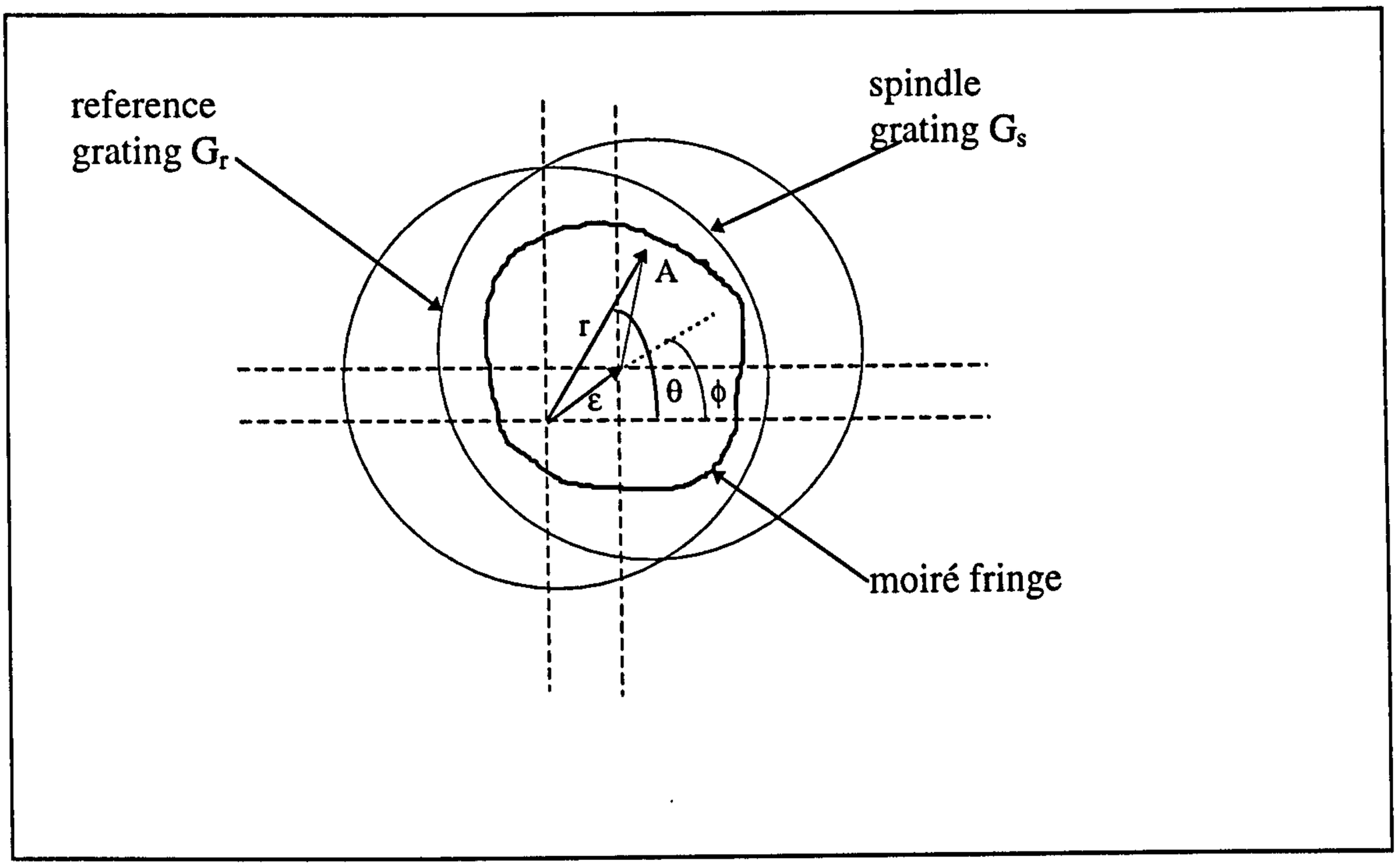


Figure 4.16: Superimposition of two circular gratings

$\mathbf{r} = r/R$: normalised parameter (r)

$\mathbf{e} = e/R$: normalised parameter (e)

r: distance from spindle grating G_r

$r_s \approx r - e\cos(\theta - \phi)$: distance from spindle grating G_s (4.19)

$$S_r = \frac{1}{2}\left(1 + \cos\frac{2\pi(N-1)}{R}r\right)$$: transmittance of grating G_r (4.20)

$$S_o = \frac{1}{2}\left(1 + \cos\frac{2\pi(N)}{R}r_s\right)$$: transmittance of grating G_s (4.21)

Equations 4.20 and 4.21 indicate that the reference grating G_r lags behind spindle grating G_s by one pitch. The measurement of the spindle radial errors is made by analysing the fringe pattern intensity profile given in equation 4.18. Using computer based interferogram analysis (ie phase unwrapping method) the moiré technique of Kim and Park (1994) gave a measurement repeatability of about 5 nm .

The Moiré fringe pattern generated from the gratings can be processed by using interferogram processing system that comprises of a CCD camera, video frame grabber/digitiser and computer as shown in figure 4.17.

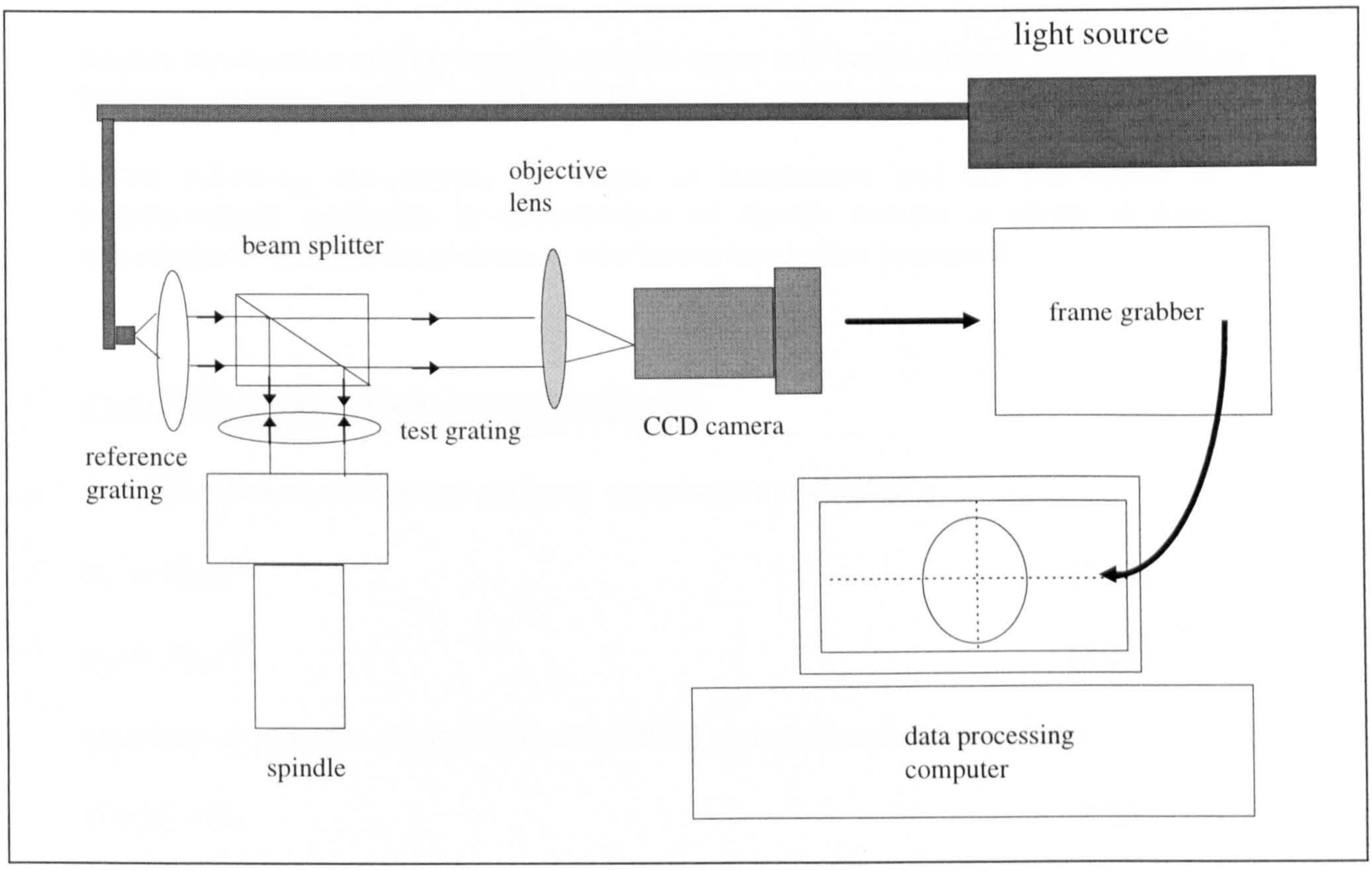


Figure 4.17: Moiré spindle measuring system after Kim & Park (1994)

4.7 INTERFEROMETRY FOR SENSING OF SPINDLE RUN-OUT

4.7.1 INTRODUCTION

The application of interferometric techniques in metrology stems from a suggestion made in the early nineteenth century, by a French physicist named Babinet, that light waves could be used as a natural standard of length (Scarr 1967). Today many measurands such as length, displacement, velocity, surface finish, heat transfer, aerodynamic properties, are measured by the use of interferometric techniques.

Typically interferometers can be classified as:

- *Wave-front dividing interferometers* such as Young's interferometer, Lloyds interferometer, Michelson's stellar interferometer and Fresnel's biprism interferometer.
- *Amplitude dividing interferometers* such as Michelson's interferometer (and its modified version by Twyman and Green, and Williams), Mach Zender interferometer, Fizeau interferometer, Sagnac interferometer, common path interferometer, Burch's scatter-plate interferometer and Pohl interferometer.

Further information may be found in suitable optics text books such as Hecht (1987) or Pedrotti and Pedrotti (1993, pp 200 - 232), for more details on these interferometers.

In the following sub-sections the theory of interference and the description of interferometers applicable to measurement of spindle run-out is given. A brief description of the sources of errors in interferometers is also presented.

4.7.2 THE THEORY OF INTERFERENCE

Interference occurs when two coherent waves with amplitudes:

$$U_1 = U_{01}e^{i\phi 1} \tag{4.22}$$

$$U_2 = U_{02}e^{i\phi 2} \tag{4.23}$$

superimpose, [see for example (Gasvik, 1987)] giving a resultant amplitude:

$$U = U_1 + U_2 \tag{4.24}$$

where, U_{01}, U_{02}: peak amplitude of the wave
$\phi 1$, $\phi 2$: phase of each wave

The intensity (I) of the superimposed waves is proportional to the *time average* of the resultant amplitude (U) squared, and is:

$$I = I_1 + I_2 + 2\sqrt{I_1 + I_2}\cos(\phi_1 - \phi_2) \tag{4.25}$$

Equation (4.25) is modified to give the general equation of the interference of two waves with phase difference: $\Delta\phi = \phi_1 - \phi_2$ and peak intensity (I_0) :

$$I = 2I_0(1 + \cos\Delta\phi) = 4I_0\cos^2\left(\frac{\Delta\phi}{2}\right) \tag{4.26}$$

When $\Delta\phi = 0, 2\pi, 4\pi, 6\pi ... n\pi$ (where n is an integer) *constructive* fringes are formed ie $I = I_{max} = I_0$. On the other hand when $\Delta\phi = \pi, 3\pi, 5\pi, 7\pi, ...(n-1)\pi$ *destructive* fringes are formed ie $I = I_{min} = 0$.

The contrast or visibility [see for example (Gasvik, 1987)] of the fringes is given as:

$$V = \frac{(I_{max} - I_{min})}{(I_{max} + I_{min})} \tag{4.27}$$

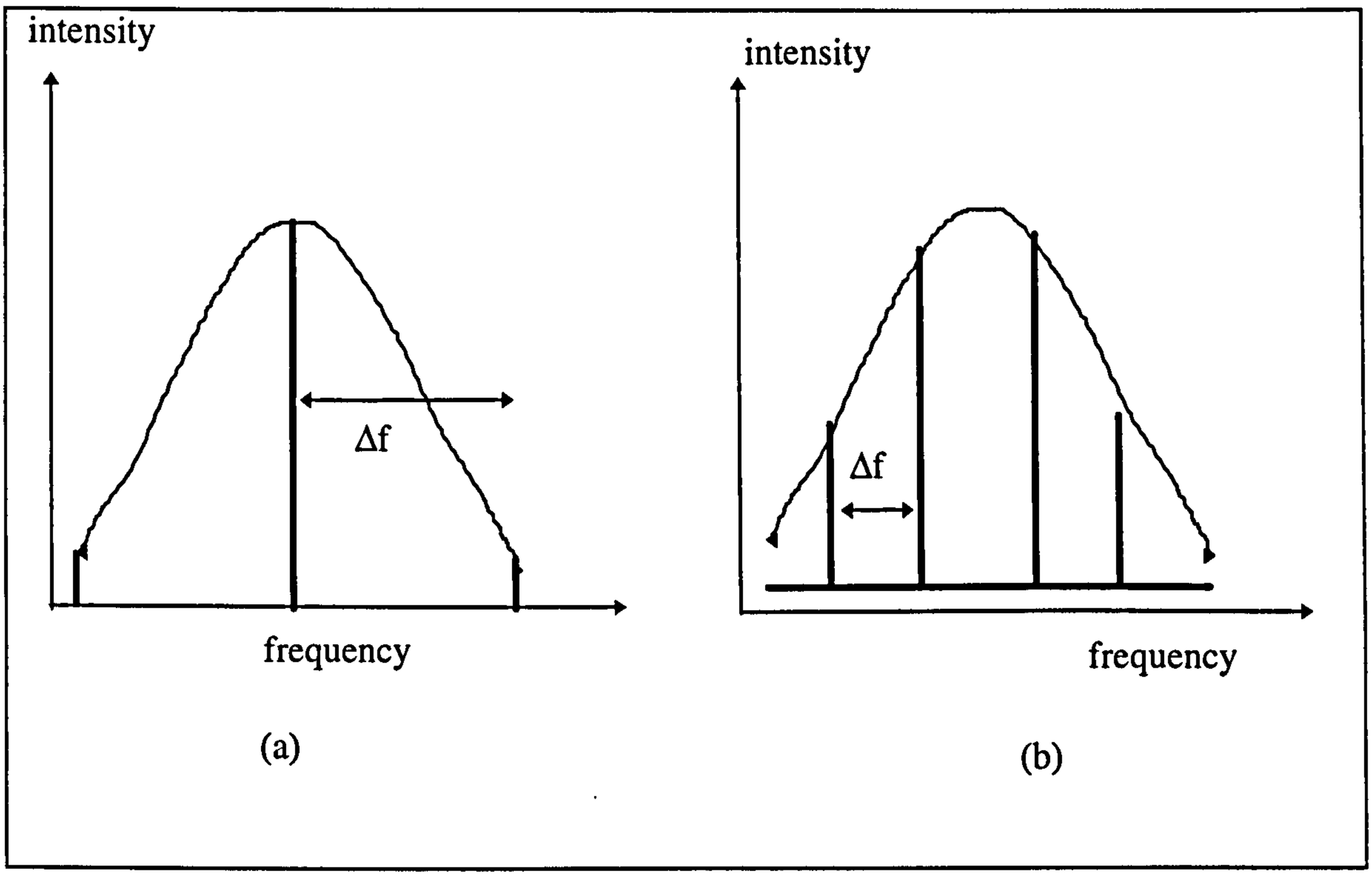


Figure 4.18: (a) Single and (b) Multi-mode spectral distribution of a He-Ne laser

4.7.3 COHERENCE OF INTERFEROMETER LASER SOURCE

Although a laser source has a well defined wavelength (or frequency) it has nevertheless a certain degree of frequency spread (Gasvik, 1995 pp104 - 105). Laser sources in interferometers normally operate in one or more distinct frequencies called modes separated by a frequency difference equal to:

$$\Delta f = \frac{c}{2L} \qquad (4.28)$$

where, c: velocity of light
L: distance between the laser mirrors

Figures 4.18a and 4.18b show the spectral distribution (gain profile) of a single and multi-mode He-Ne laser source.

A laser source is usually characterised by an average emission life time called *coherence time* (τ_0); which is proportional to the natural width ($\Delta\lambda$) of the source spectral line. The monochromacity of a laser source improves with higher coherence time. The *coherence length* (l_t) of the source is equal to:

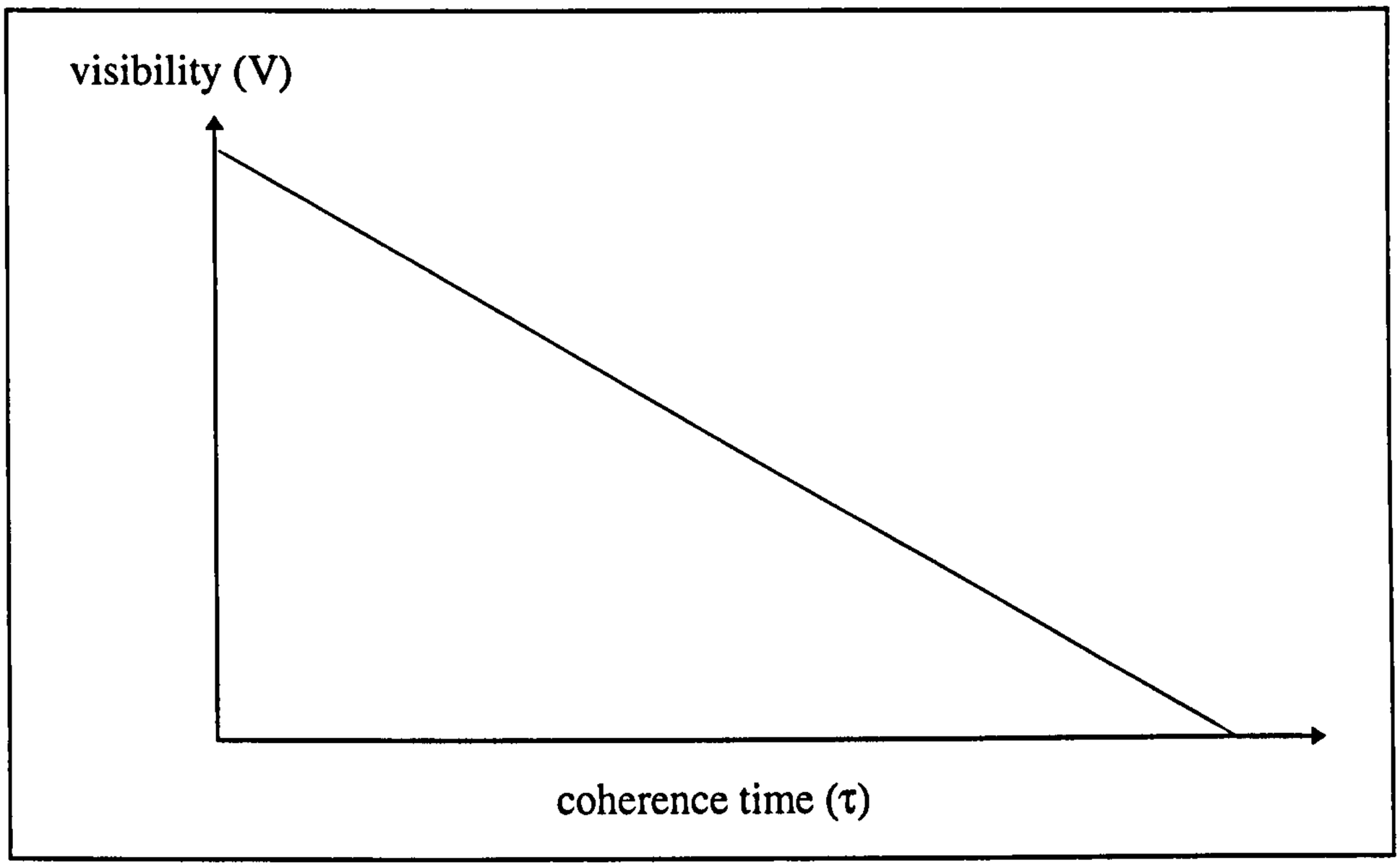


Figure 4.19: Graph of fringe visibility against coherence time

$$c\tau_0 = \frac{c}{\Delta f} \qquad (4.29)$$

and is a measure of the length over which the laser source is coherent. Modifying equation (4.29) (Pedrotti & Pedrotti, 1993, pp247 - 262):

$$l_t = \frac{\lambda^2}{\Delta\lambda} \qquad (4.30)$$

Hence, from equation (4.38) the natural line width is:

$$\Delta\lambda = \frac{\lambda^2}{l_t} \qquad (4.31)$$

The coherence length of (interferometer) a laser source determines its measuring range. Typical coherence length of a muti-mode He-Ne laser such as the one used in the Fizeau interferometer for this project, is approximately 0.2 m compared with coherence length of 1 μm for a white light source (Wilson & Hawkes, 1989 pp216 - 236). Single mode He-Ne lasers are known to have coherence lengths up to 1000 m (Wilson & Hawkes, 1989 pp216 -236).

Temporal coherence of a source influences the degree of visibility of an interferogram (Pedrotti & Pedrotti, 1993 pp247 - 262). Figure 2.19 shows the relationship between fringe visibility (V) and coherence time (τ_0). Special cases of the influence of temporal coherence and fringe visibility are:

complete incoherence: $\tau \rightarrow \tau_o$: $V = 0$

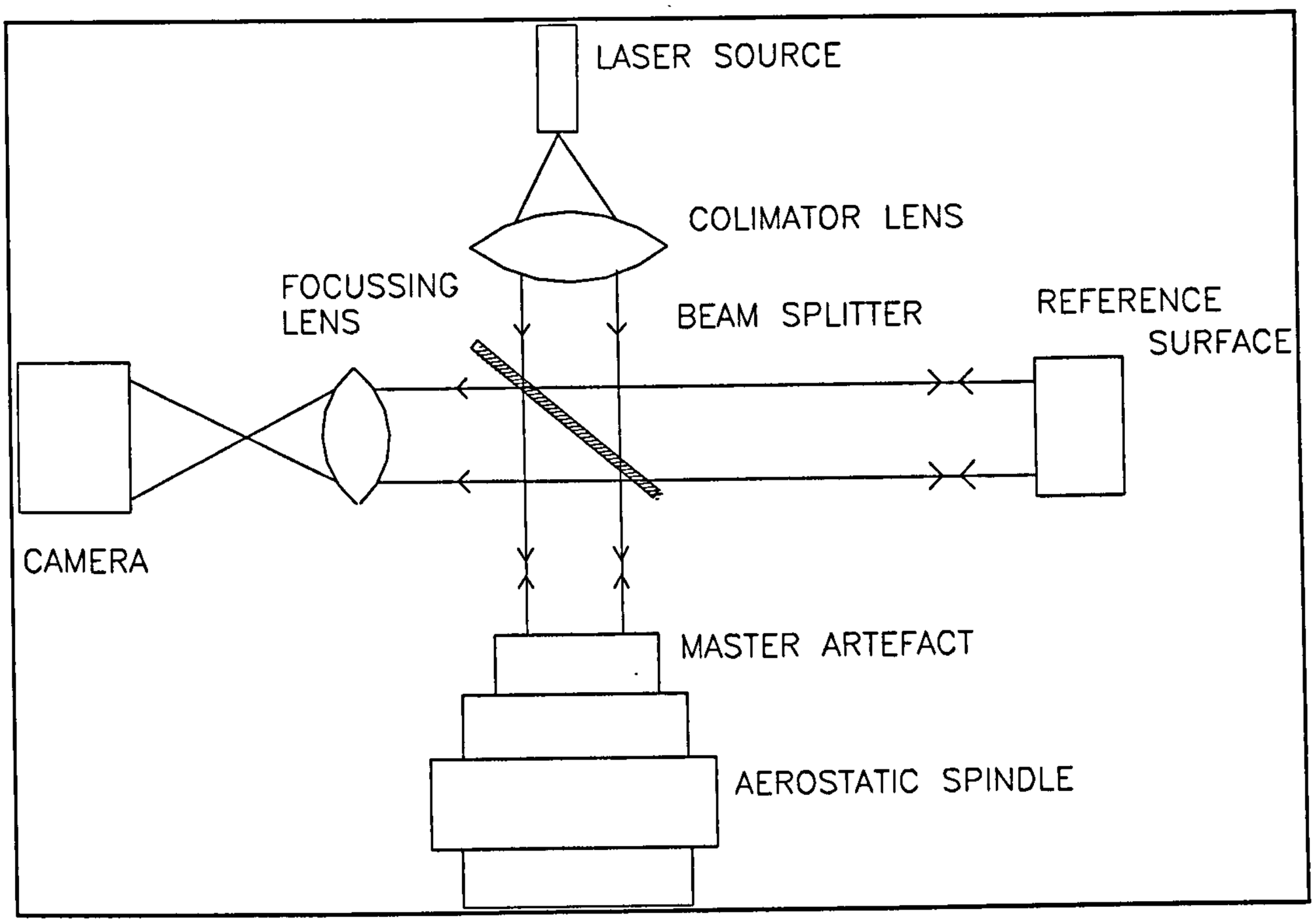


Figure 4.20 : Configuration of Michelson interferometer

complete coherence: $\tau = 0$: $V = 1$
Partial coherence: $0 < \tau < \tau_o$: $1 > V > 0$

4.7.4 INTERFEROMETERS FOR MEASURING SPINDLE RUN-OUT

Typical interferometer configurations that can be used for measuring spindle run-out are:

- Michelson interferometer
- Williams interferometer
- Common path interferometer
- Mach Zehnder interferometer
- Fizeau Interferometer.

4.7.4.1 MICHELSON INTERFEROMETER

Figure 4.20 shows the configuration of a Michelson interferometer when used to analyse the spindle error motions of a (vertical) aerostatic spindle. The interferometer has a laser source that is collimated to a beam splitter where part of the light is reflected to a reference surface (Slocum, 1992 pp181 -186), while the other part of the light travels to the surface of a *master* artefact placed on the spindle. Each (partial) beam is reflected

back from the respective surfaces to re-combine at the beam splitter, and the path difference produces interference fringes that can be detected by a by a photodetector.

The optical path difference between the two beams, is due to any axial and/or tilt error motions of the spindle. For instance an axial error of d on the spindle will result in a path difference of:

$$\delta = 2\,d \tag{4.32}$$

and phase difference:

$$\Delta\phi = k\,\delta = \frac{2\pi}{\lambda}.2d \tag{4.33}$$

Substituting equation (4.33) into equation (4.26) gives the intensity equation for a Michelson interferometer:

$$I = 4I_0\cos^2\left(\frac{2\pi d}{\lambda}\right) \tag{4.34}$$

The spindle run-out can then be obtained from the intensity profile of the formed interferogram at angular positions sampled as the spindle rotates.

4.7.4.2 WILLIAMS INTERFEROMETER

A Williams interferometer can be used to simultaneously measure all the five DOF error motions of a spindle. The interferometer is a modification of a Tywman-Green interferometer [for example see, Houston et al, 1967)] using spherical mirrors instead of plane mirrors as the *test* and *reference* mirror surfaces. Figure 4.21 shows a schematic of a Williams interferometer used for measuring spindle run-out. The instrument incorporates a laser source whose wavefront is divided in amplitude by beam-splitter (BS). The divided wavefronts that emerge from BS travel towards the individual test and reference mirror surfaces: (MT) and (MR) respectively. After reflection, light from MT and MR combined at BS forming interference fringes, which can be detected with the aid of camera (D).

4.7.4.3 MACH ZEHNDER INTERFEROMETER

Wilkie and Fisher (1963), describe the Mach-Zenhnder interferometer, an instrument primarily used for the study of aerodynamic problems. Figure 4.22 illustrates a schematic of how it might be used to analyse spindle error motions of a horizontal aerostatic spindle. The principle of operation of this interferometer involves dividing the source wave into a reference beam and a test beam by a beam-splitter (the first). The test beam is used to measure the spindle run-out, using a second beam splitter, the beams meet (interfere) to form a fringe pattern, which can be detected by eye or by a camera.

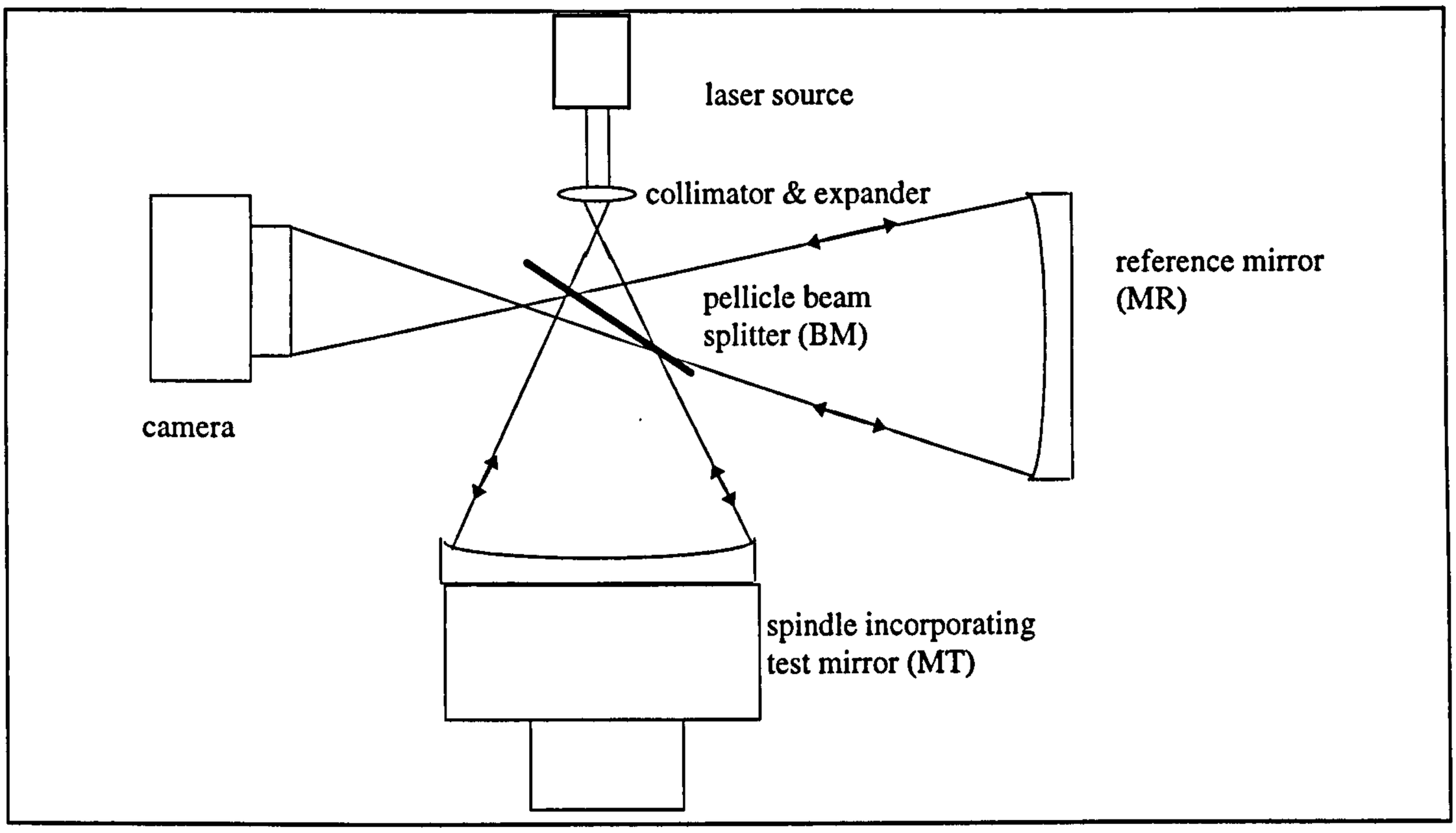


Figure 2.21: Configuration of Williams interferometer Houston et al (1967)

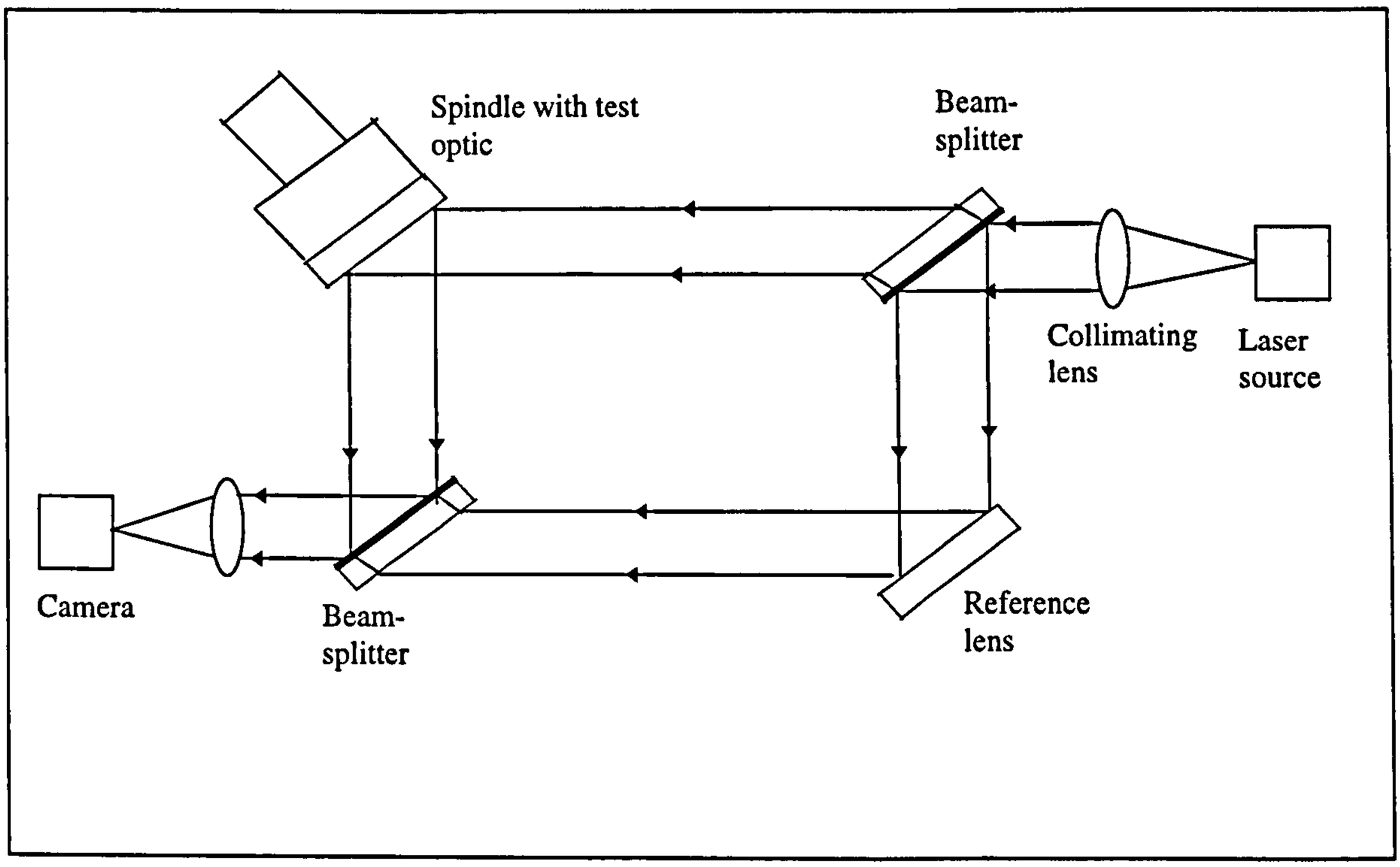


Figure 4.22 : Configuration of the Mach Zehnder interferometer

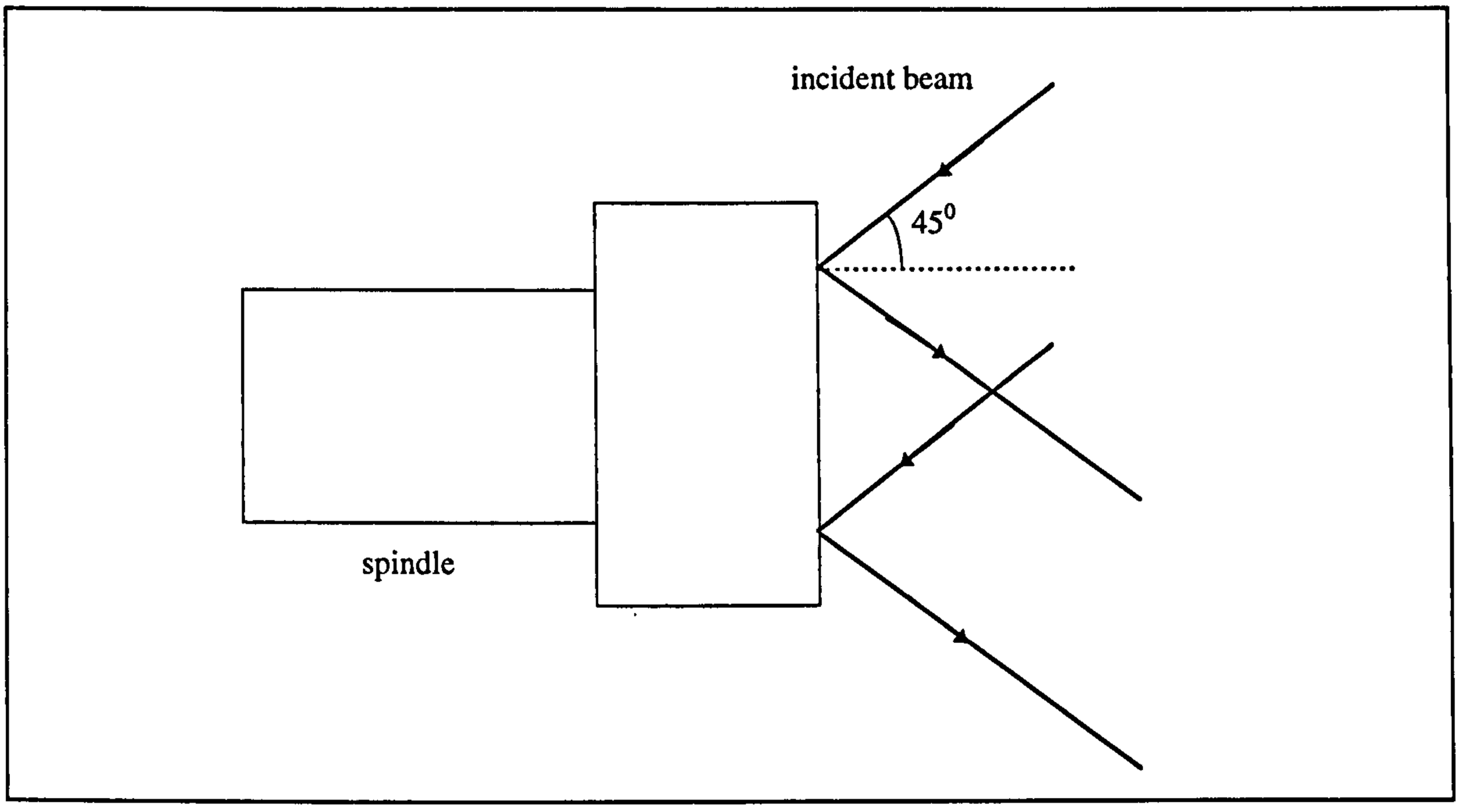


Figure 4.23 : Oblique nature of interferometer normal to spindle plane

The optical path difference (eg due an axial error of d) is:

$$\delta = \frac{2d}{\cos 45} \qquad (4.35)$$

The existence of the cosine term is due to the oblique nature of the beam incident on the surface of the master artefact placed on the spindle, as shown in figure 4.23.

The phase difference is :

$$\Delta\phi = k\,\delta = \frac{4\pi d}{\lambda \cos 45} \qquad (4.36)$$

which gives an intensity equation of :

$$I = 4 I_0 \cos^2\left(\frac{2\pi d}{\lambda \cos 45}\right) \qquad (4.37)$$

4.7.4.4 COMMON PATH INTERFEROMETER

Application of a *common path* interferometer will reduce the influence of ambient temperature, pressure and humidity variations on the optical path length (Slocum 1992, pp192-202). According to Dyson (1970), the principle of a common path interferometer involves sending both beams (ie interferometer reference and measured beams) along the same general path through the instrument in such a way that they are affected differently by aberrations or imperfections of the optics. This thus makes the interferometer suitable for the inspection of large optics.

There are a number of common-path interferometer configurations used in metrology. The configuration which can be used to measure spindle run-out is the *scatter-plate* interferometer developed by Burch (1953).

The arrangement of a scatter-plate interferometer for monitoring spindle performance is shown in figure 4.24. Light from a laser source passes through a beam-splitter (BS) and a scatter-plate where the primary amplitude division of two beams occurs (Horne, 1983). One beam traverses the full clear aperture of the reflecting optic placed on the spindle and serves as the *measured beam.* While a second traverses a small portion of the optic, and serves as the *reference wavefront.* Interference between these two beams produces a fringe pattern, which can be used to quantify spindle error motions. This type of interferometer is capable of detecting the spindle orthogonal tilts (E_x, E_y) and radial error motions (d_x, d_y).

4.7.4.5 FIZEAU INTERFEROMETER

Fizeau interference fringes can be observed by illuminating an optical flat with a collimated laser beam as shown in figure 4.25. In this arrangement the ray ABC is reflected from the back surface of the optical flat along the path CEF, while the beam CD is reflected from the surface of the workpiece along the path DGHI . The ray DGHI is longer than CEF by an optical path difference of CDG. When CDG is equal to one or a whole number of wavelengths, both the interfering beams will be in phase (Erwin, 1967) and *constructive* interference will occur. On the other hand if CDG is equal to one and half or generally half and a whole number of wavelengths, both beams will be out of phase, hence *destructive* interference will occur.

The Fizeau interferometer is modification of the optical flat arrangement (which is still used today for the calibration of slip gauges). Figure 4.26 shows the Fizeau interferometer (Gee et al, 1988) used in this work, it incorporates a coherent collimated source (normally a He-Ne laser source), and a test optical surface, and a reference surface provided by an optical flat on the spindle. Figure 4.27 depicts the complete interferometric instrument incorporating tilt control servos and computer control.

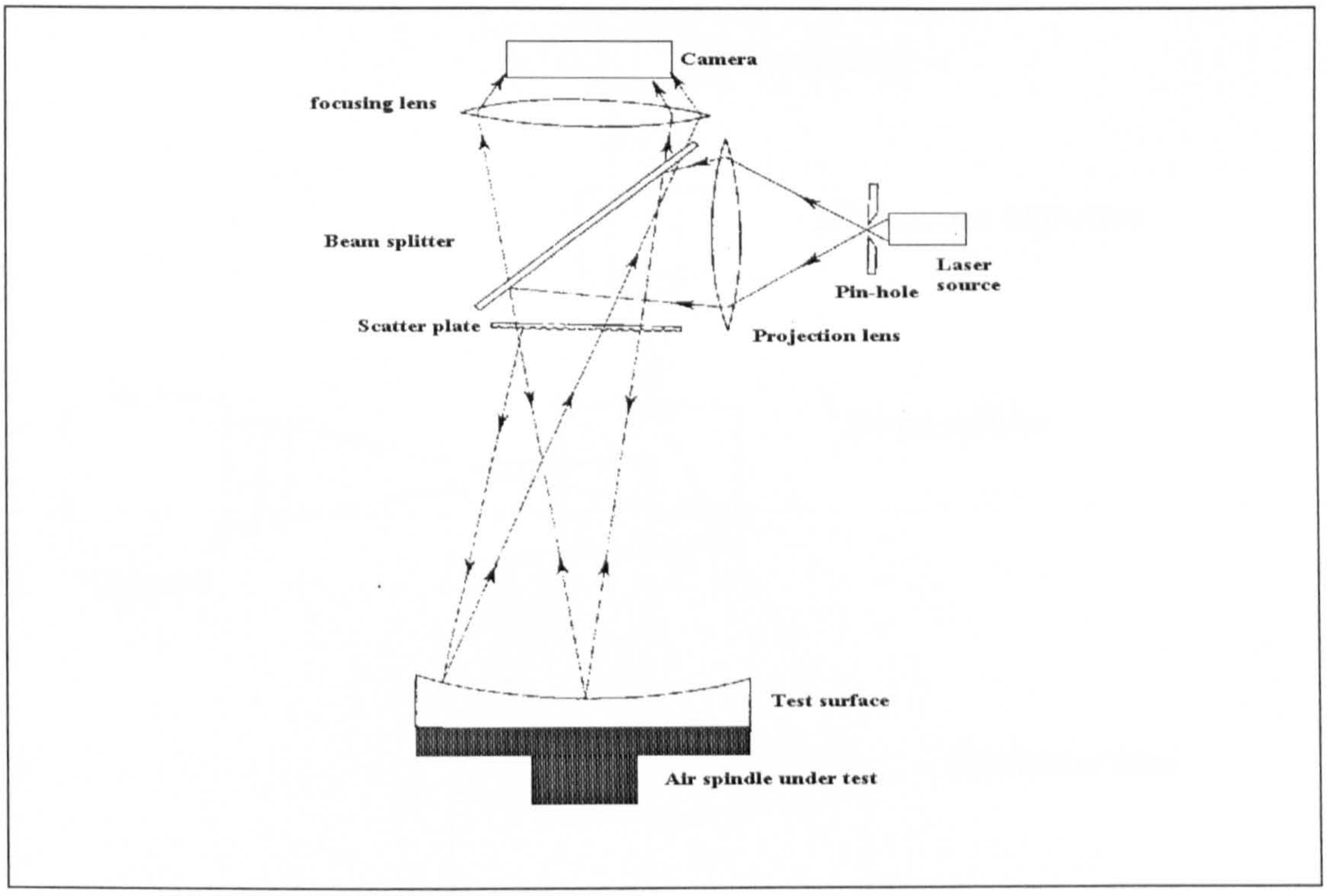


Figure 4.24 : Configuration of the Common path interferometer (after Horne, 1983)

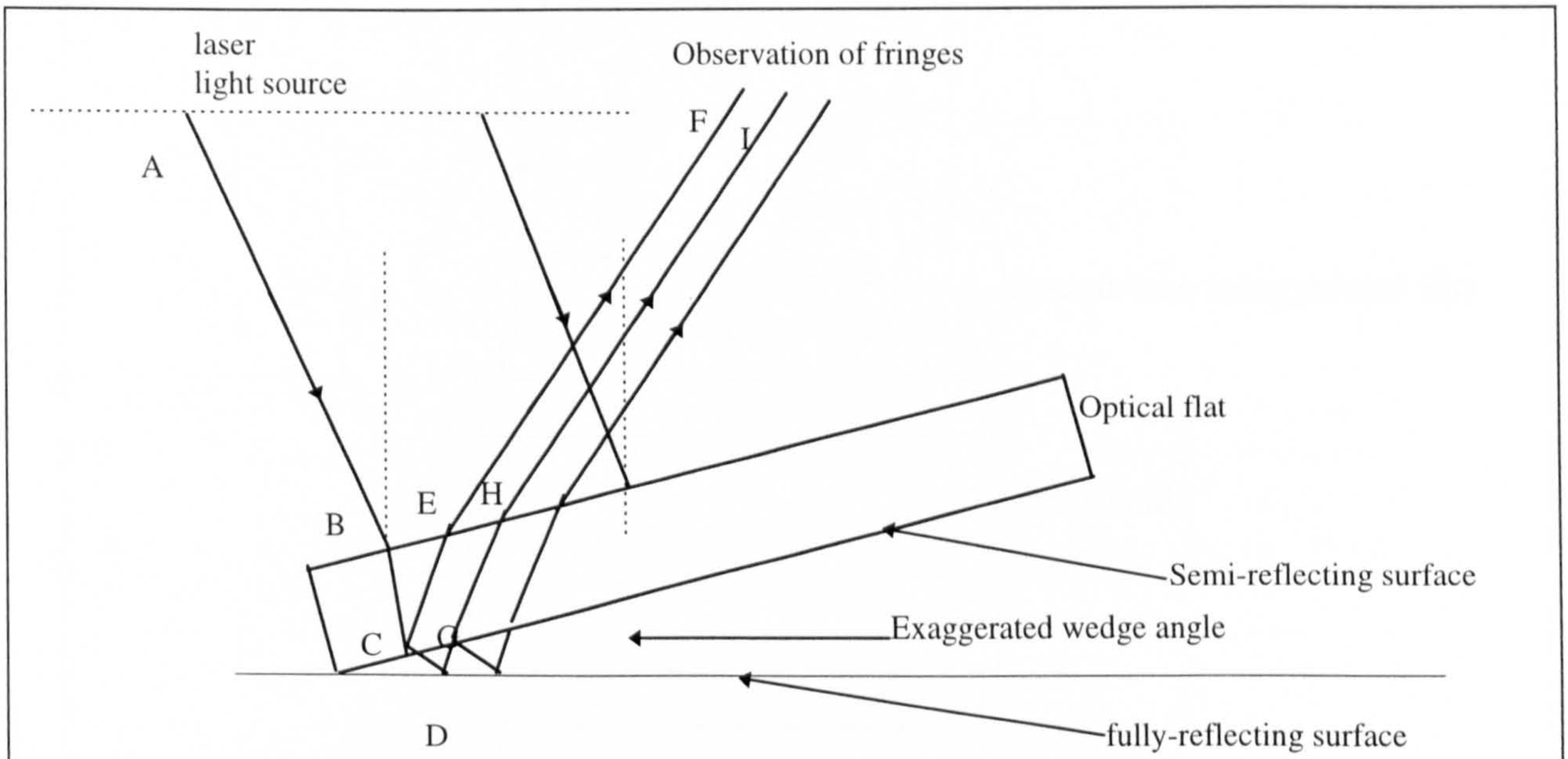


Figure 4.25 : Principle of Fizeau fringe formation (after Erwin, 1967)

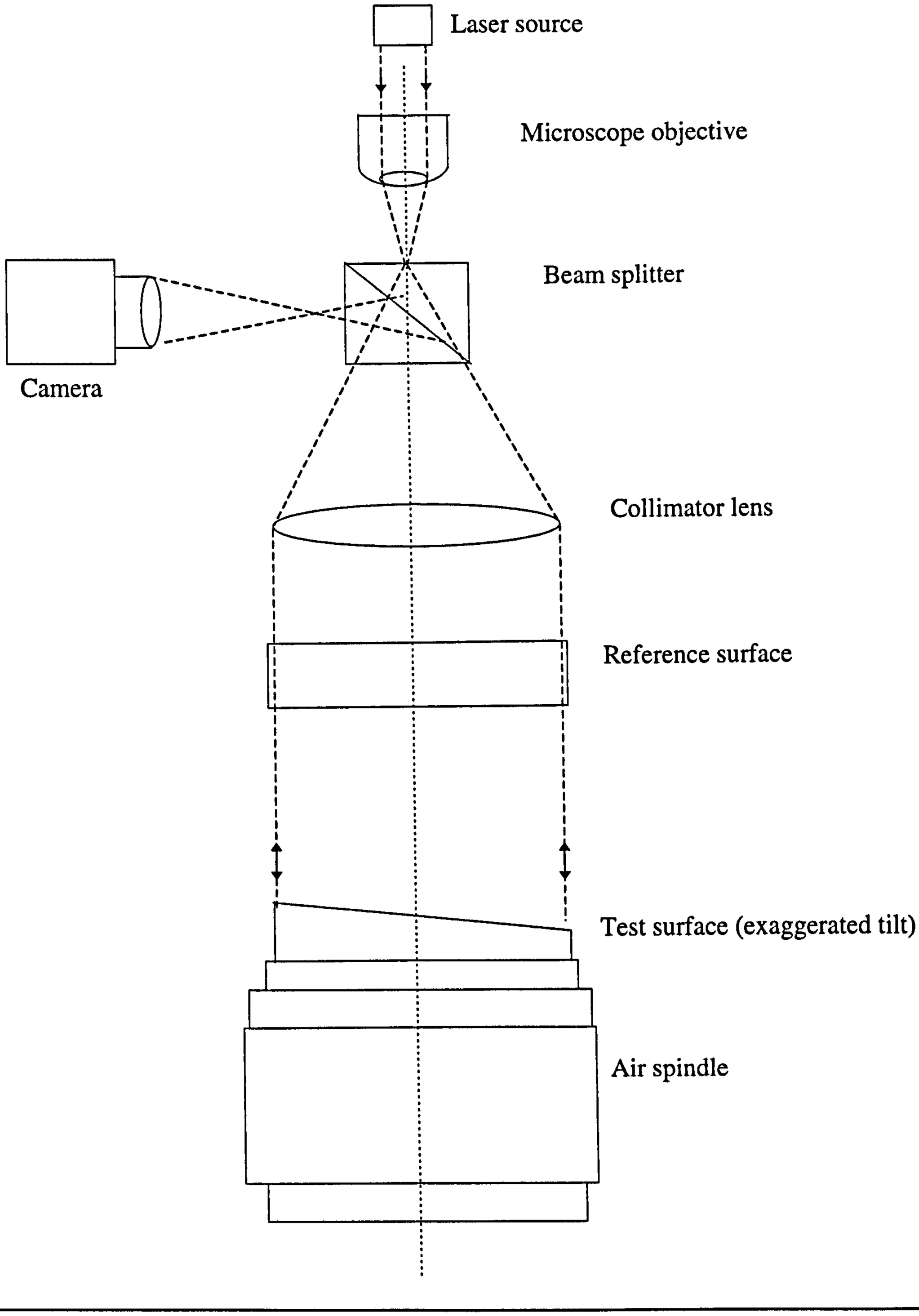


Figure 4.26: Fizeau interfermeter set-up for measuring spindle run-out after Gee et al (1988)

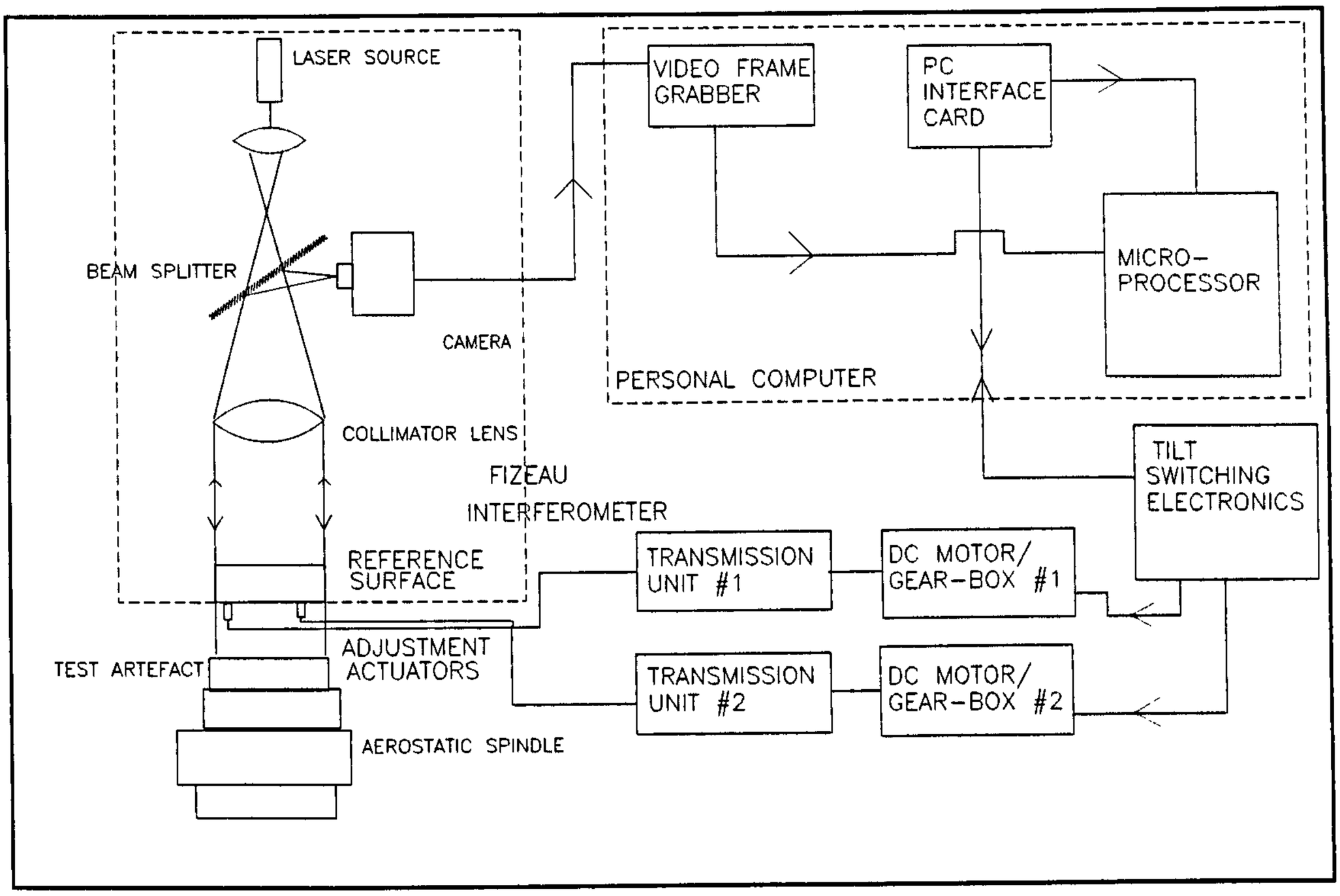


Figure 4.27 : Configuration of a Fizeau interferometer when used to measure spindle run-out

4.7.5 SOURCES OF ERRORS IN INTERFEROMETERS

According to Slocum (1992 pp174 - 176) the error resulting from an interferometric measurement is a function of medium refractive index (n), light source wavelength (λ), detector electronic rms (root mean square) noise and sensor misalignment error. Major errors in interferometers and ways of reducing these problems are described in the following sub-sections.

4.7.5.1 REFRACTIVE INDEX ERRORS

The accuracy of an interferometer measurement is usually affected by the instability of its *dead-path* due to changes in the refractive index (n). The dead-path is path of the interferometer optical path length which is sensitive to environmental variations, the Fizeau interferometer used in this work had a dead-path of approximately 300 mm. Environmental variations in temperature, pressure, humidity and gas composition affect the refractive index of the light source in the measurement area

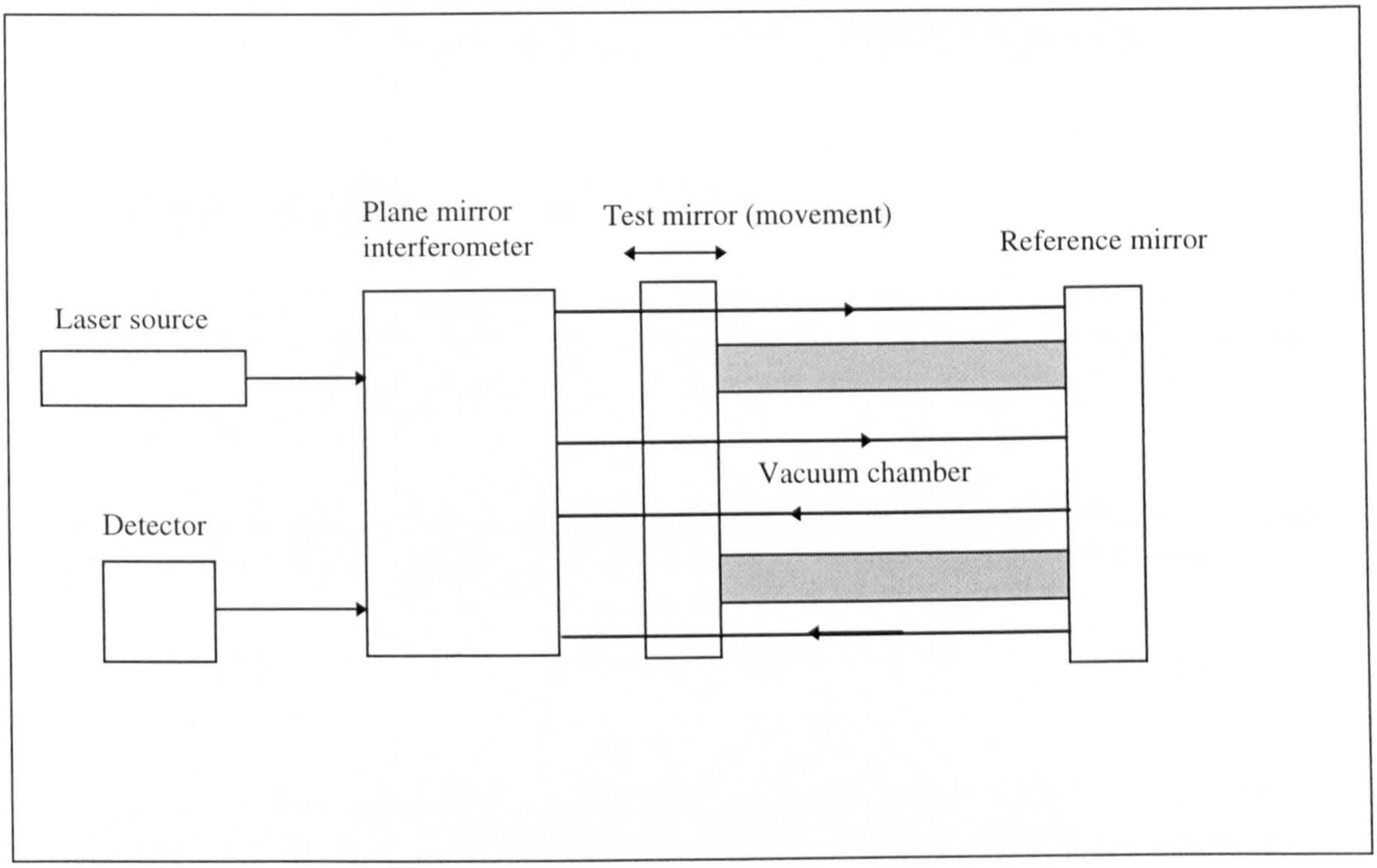


Figure 4.28: Schematic of a Refractometer (after Slocum 1992, pp192-202)

(Slocum, 1992 pp192 -202). The relation between refractive index and the variations in temperature, humidity and pressure is given by Schellekens (1986):

$$n = \frac{2.879294 \times 10^{-9}(1 + 0.54 \times 10^{-6})(C - 300)P}{1 + 0.003671 \times T} - 0.42065 \times 10^{-9} \times F + 1 \qquad (4.38)$$

where, T: ambient temperature (degrees)
P: ambient pressure (Pa)
F: water vapour pressure (Pa)
C: CO_2 content (PPM)

The partial derivatives of equation 4.45 can be used to estimate the influence of the respective changes in C, P, F and T on n (Slocum, 1992 pp192 - 202), for specified nominal values of humidity (carbon content in air), pressure and temperature.

An effective way to measure the influence of these error contributions on the refractive index is by the application of a *refractometer* (Hosoe and Tanara 1995). Figure 4.28 depicts a schematic of a refractometer, the instrument consist of a vacuum chamber, measurement mirror, reference mirror, interferometer (ie used to measure for measuring the refractive index) and ancillary components for the vacuum system.

Hosoe and Tanara (1995) have developed a low cost refractometer which has demonstrated improvements in displacement drift due to changes in refractive index of 4.8 nm and 1.2 nm for atmospheric and vacuum conditions respectively.

4.7.5.2 INSTABILITY OF LASER SOURCE

Instability of laser frequency due to thermal drifts causes the effect of *Doppler broadening*. This effect is due to the random nature of laser (source) atoms, causing it to emit waveforms at different resonant frequencies, resulting in a *broadening* of the frequency bandwidth (Δf).

Wilson and Hawkes (1989), suggested that this effect applies to a collection of atoms emitting at an optical frequency (f_o), so that the observed frequency is given by:

$$f_m = f_o\left(1 \pm \frac{v_x}{c}\right) \qquad (4.39)$$

where v_x is the component of the velocity of the atom along the direction of observation (assuming v_x is much lower than the velocity of light c). There are a number of ways of lowering or *narrowing* the mode shifted bandwidth frequency of a laser source, these include:

- Varying the laser cavity length by heating or by a piezoelectric actuator until the value of f_m in equation (4.39) is equal to f_o (Wilson & Hawkes, 1989 pp216 -236)

- In a two-mode laser source, any thermal drift in the cavity will cause the amplitude/intensity of one mode to increase and the other to decrease. In this condition, the laser is stabilised by separating the modes using a polarising beam-splitter and separate detectors to monitor the intensity signal of each polarisation (Williams 1993).The difference in the intensity signal is used in a feedback loop to change the length of the laser cavity by servoing it with a heater or piezoelectric actuator. This technique gives a stability of 1 part in 10^9 when used in commercial He-Ne lasers.

- According to Williams (1993) one of the most accurate and well-established techniques for providing long term stability is to reference the laser to saturated spectra in diatomic molecular iodine vapour (Hanes & Dahlstrom, 1969).This technique has a stability of about 1 part in 10^{12}.

4.7.5.3 INTERFEROMETER MISALIGNMENT ERROR

Interferometer misalignment adds considerably to the total measurement error of an interferometer. Figure 3.29 shows an interferometer mis-aligned with respect to the required measurement axis of motion (axial displacement) in the case of a spindle. As a result of this misalignment, the sensor will measure a spurious component of the

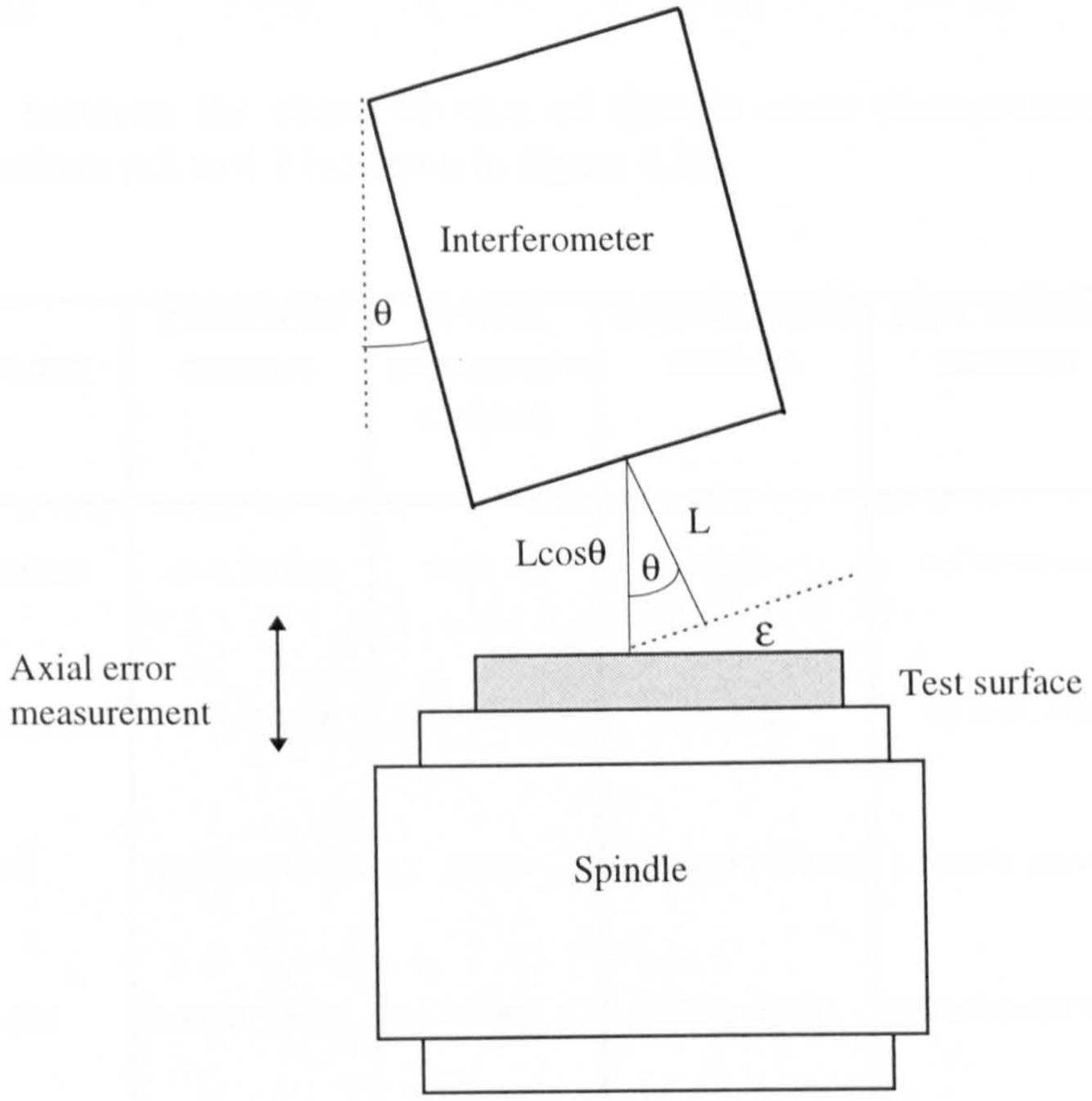


Figure 4.29: Influence of cosine error in interferometric spindle error motion measurement

spindle axial displacement (d), usually referred to as a *cosine* error, and whose magnitude is:

$$\varepsilon = L(1 - \cos\theta) \qquad (4.40)$$

4.7.5.4 DETECTOR ERROR

The amount of electrical noise in a heterodyne detector (eg phase detector) corresponds to an rms phase noise of about 1 part in 10^3 (Slocum 1992 pp192 - 202). In this project a camera was used to detect the interferogram. This photodetector has errors due to non-linearity of photoelements, influence of *dark current*, non-uniform ambient illumination effects, and fringe pattern minima or maxima measurement error (see section 6.5 for further details on the accuracy of fringe peak detection).

4.8 COMPARISON BETWEEN SPINDLE RUN-OUT MEASUREMENT TECHNIQUES

A comparison between the characteristics of spindle error measurement techniques described in sections 4.2 to 4.7 is shown in figure 4.30.

SENSING TECHNIQUE	CAPACITIVE SENSING	OPTICAL DIFFRACTION SENSING	OPTICAL FIBRE SENSOR	EDDY CURRENT SENSING	(THIS PROJECT) FIZEAU INTERFEROMETRIC SYSTEM
MEASURING RANGE	+/- 0.25 mm	1000 nm	5 mm	0.5 to 50 mm	0.03 mm
FREQUENCY RESPONSE	30 - 40 kHz	10 - 60 MHz	10 kHz	up to 50 kHz	30 Hz (typical frame rate)
RESOLUTION	less than 10 nm	25 nm	0.10 nm to 100 nm	0.1 nm to 3000 nm	less than 30 nm
REPEATABILITY	about 10 nm	60 nm	0.05% of FSD	200 nm to 6000 nm	about 60 nm
ENVIRONMENTAL EFFECTS ON ACCURACY	very critical	not critical	not very critical	not very critical	very critical

Figure 4.30 : Comparisons between the spindle measurement techniques

CHAPTER 5: INSTRUMENTAL CONSIDERATIONS

5.1 INTRODUCTION

An interferometric spindle metrology system has been developed in this work. In this chapter, a description of the hardware components of this instrument is presented. Description of the interferometer Vidicon camera and its limitations is given, including a detailed explanation of an automatic interferometer adjustment servo and spindle position measurement sensor.

The interferometric spindle analysis system, consists of the following hardware:

- Fizeau interferometer type PTI manufactured by the Zygo company*
- Personal Computer
- Video frame grabber card
- PC interface card
- Resolver and resolver-to-digital converter (RDC)
- Interferometer auto-adjustment servo-mechanism

5.2 FIZEAU INTERFEROMETER

The Fizeau interferometer used in this project is the Zygo model PTI (Production Test Interferometer). The interferometer system consists of:

- Fizeau interferometer mainframe (see section 4.7.4.5)
- Monochrome (Vidicon) camera unit (see section 5.3.1)
- Video monitor
- An automatic 2-axis servo adjustment mechanism (see section 5.5)

Figure 5.1 shows a photograph of the Fizeau interferometer mainframe. Figure 5.2 illustrates how the interferometer was attached vertically to a research machine incorporating an aerostatic spindle.

---

* Zygo Corporation, Laurel Brook Road, P. O Box 448, Middlefield, CT 06455, USA

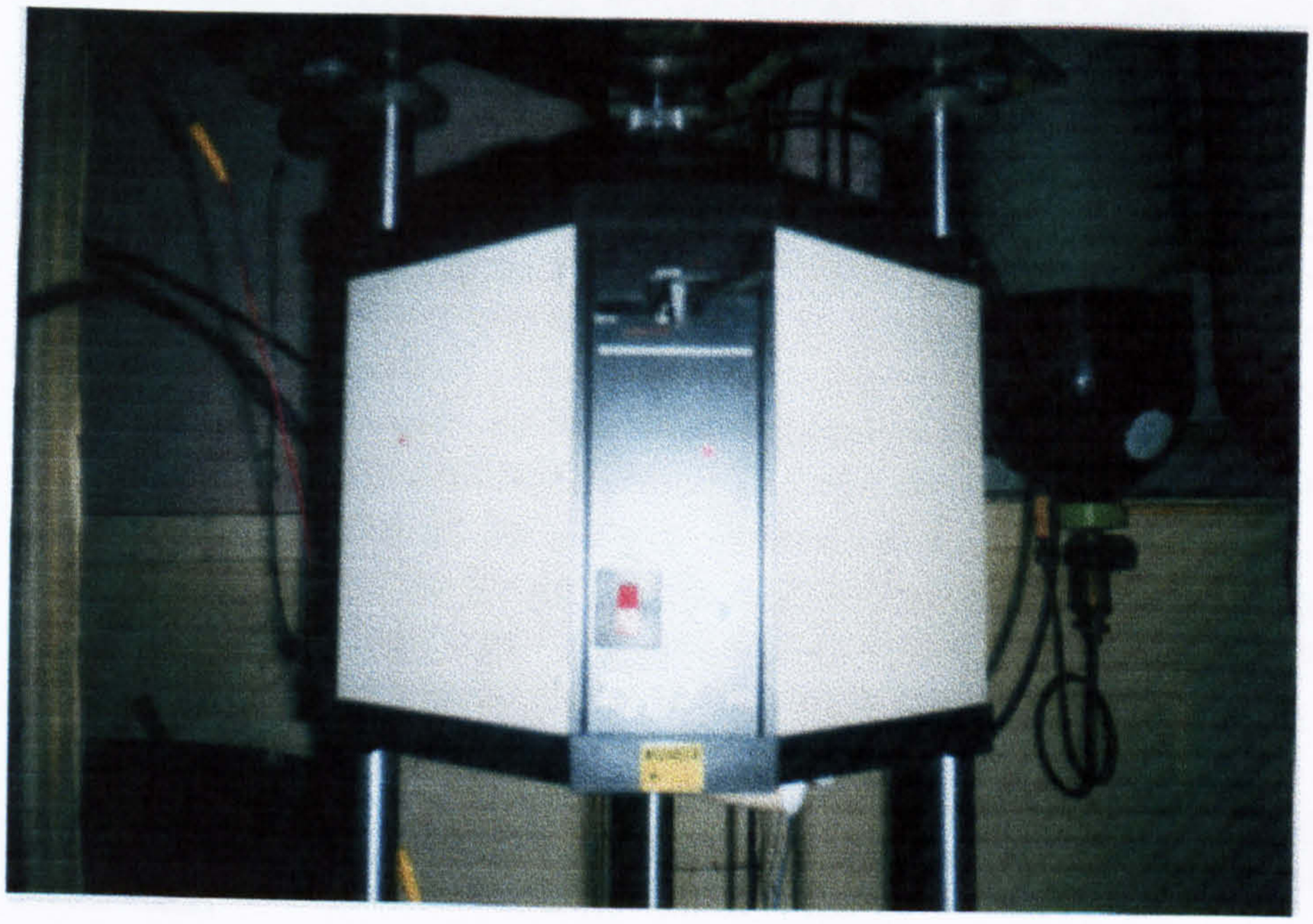


Figure 5.1 Interferometer mainframe

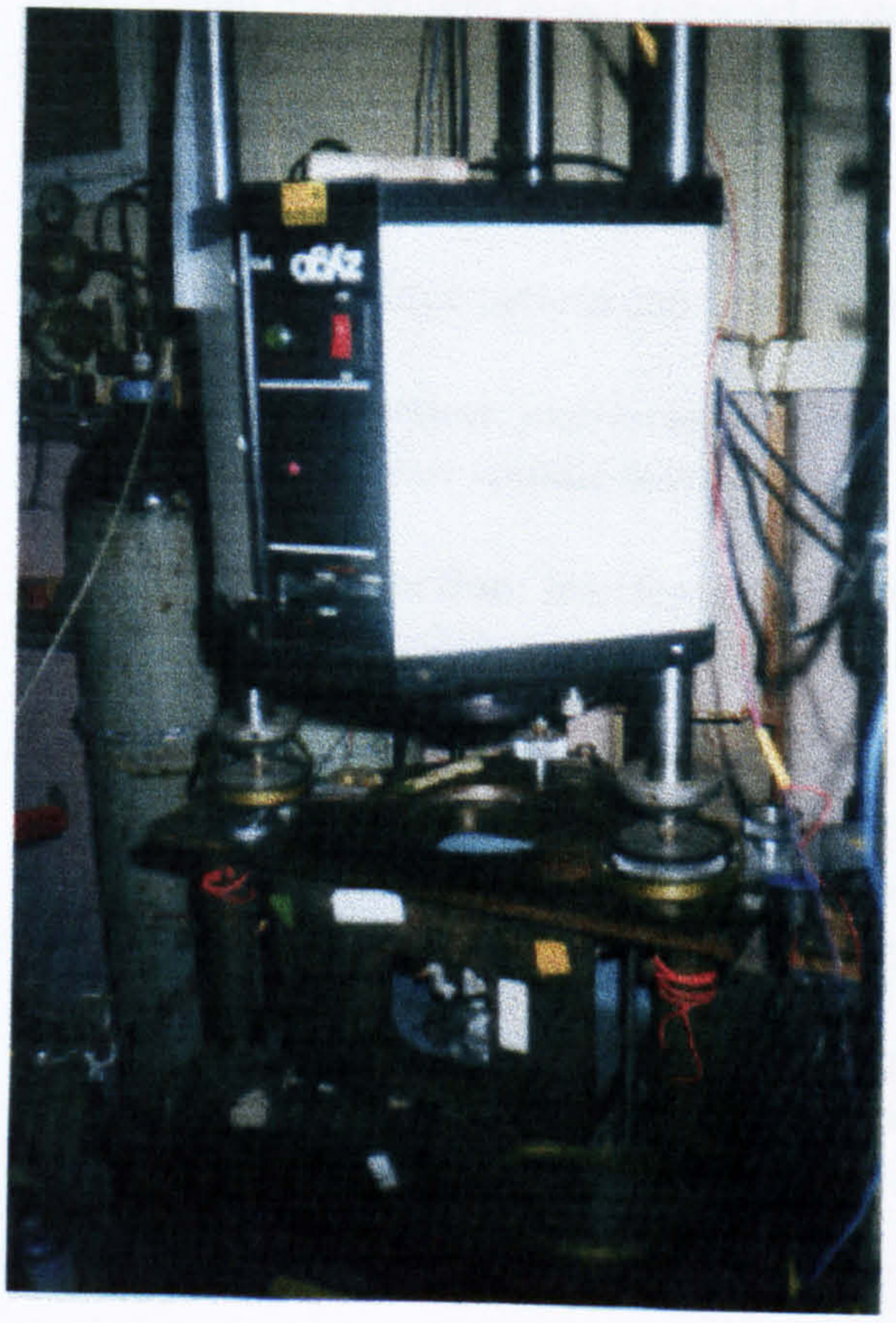


Figure 5.2: Attachment of the interferometer spindle machine

5.3 VIDEO DETECTION OF INTERFEROMETER SIGNAL

5.3.1 APPLICATION OF VIDICON CAMERA

A vidicon camera is incorporated within the PTI interferometer to detect interference fringes formed in real-time. Figure 5.3 shows a schematic of a Vidicon camera, it consists of an electron gun which produces an electron beam (A) through aperture (B). The beam scans a target surface incorporating photoresistive elements (C). Interferograms formed during the experiment are detected by the camera photoresistive (semiconductor) elements which are scanned line by line at a rate of 64μs. The resistance of the camera photoresistive elements is proportional to the detected intensity of the fringe pattern, and subsequently proportional to the charging current. The charging current signal is used to recreate the fringe pattern in real-time at rate of 25 frames a second when the camera is connected to a monitor.

5.3.2 INACCURACIES IN VIDEO DETECTION WHEN USING A VIDICON CAMERA

One of the main considerations that has to be made during the interferogram processing stage is how to capture and store the fringe pattern data in a duration shorter than the changes being detected in the fringe pattern due to spindle run out. In the case at hand of rotating fringes, a camera is required to detect the fringe phase or spatial frequency changes instantaneously. During a single frame detection, changes in the pattern must be detected equally by all the elements of the camera.

Detection of the fringe pattern can achieved by two methods, these are:

Synchronous detection: interferograms of the spindle run-out are captured after an angular interval in one spindle revolution.

Asynchronous detection: interferograms of the spindle run-out are captured at specified angular positions over a number of spindle revolutions.

Typically, fringe detection ambiguities may occur due to *aliasing*. Aliasing causes the sampled interferogram to appear distorted or swivelled. The distortion of the interferogram is due to the make-up of the sampled frame, which is composed of sub-frames captured as the spindle rotates through discrete angular positions during the 40 ms frame sample period. One way of describing the swivelling effect of a synchronously detected fringe pattern is to consider a hypothetical case of a spindle revolving at 2 revs per second, where a vidicon camera is used to detect the formed interferograms at a one degreee angular interval. The following data describes this condition:

1 frame capture takes 40 ms (25 frames/s)
1 frame contains 625 lines
1 line scan takes 64 μs

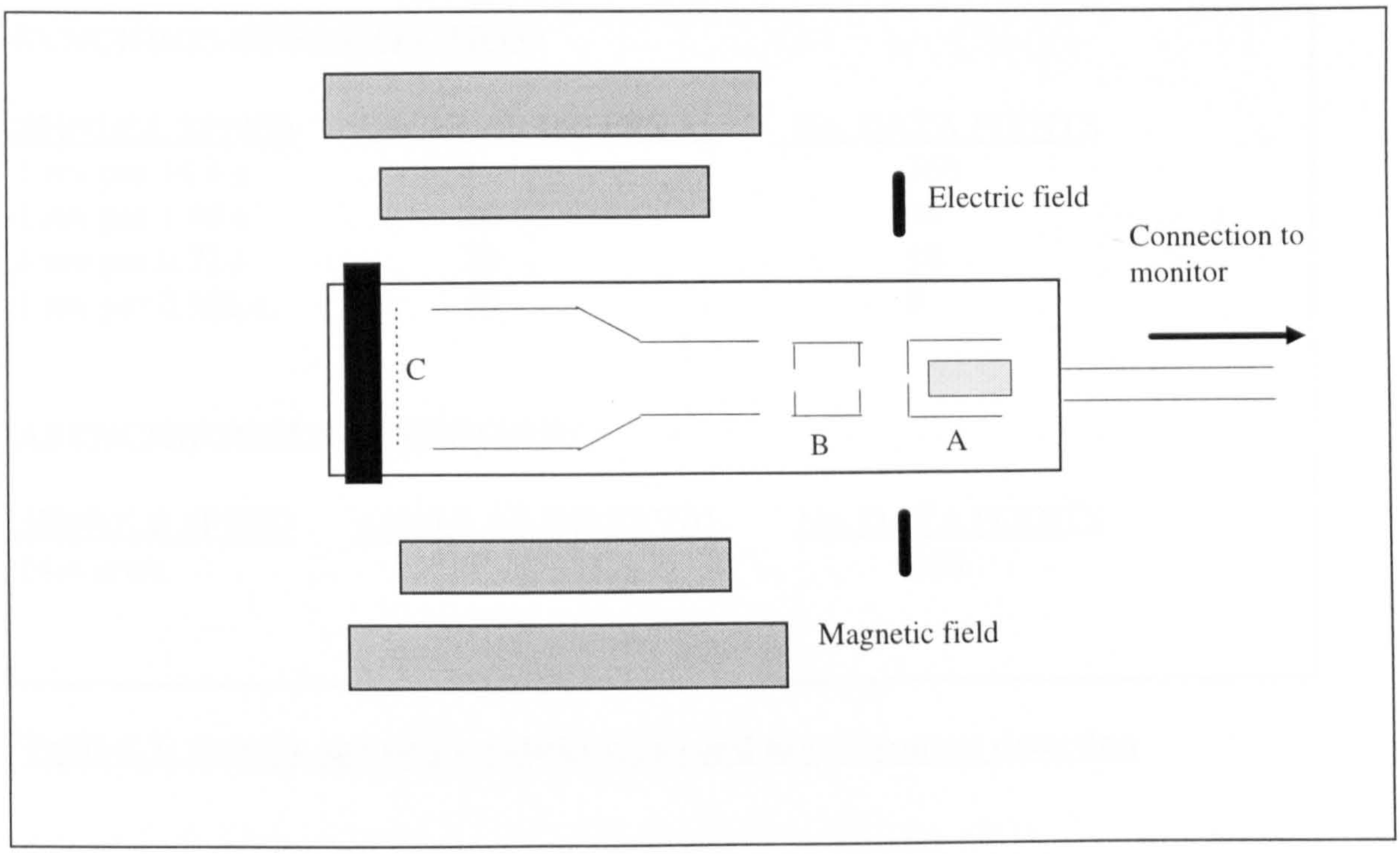


Figure 5.3: Schematic of Vidicon camera after Gasvik (1995 pp109 - 114)

2 revs per secs is 0.5 s or 500 ms
1 degree angular position takes 1.38 ms
360 degree angular position takes 500 ms

From the above data, in 40 ms the spindle would have moved to an angular position of approximately 28.9 degrees meaning that the *resultant* frame is part of the *individual* frames detected as the spindle moved through the 1,2,3, ...,28.9 degrees. This implies that, at every one degree interval, the resultant frame contains 21.63 lines of each of the individual frames to give a single frame of 625 lines, and this is what causes the interferogram to appear swivelled.

In order to capture an error-free rotating fringe pattern at one degree angular intervals, the spindle must revolve at a speed less than or equal to 1 rev per 14 seconds (synchronous fringe detection) or 14 revs per second (asynchronous fringe detection). Table 5.1 below gives a summary of the relationship between spindle speed, (detection) angular interval, and number of data points per polar chart for both synchronous and asynchronous detection:

SYNCHRONOUS DETECTION:

SPINDLE SPEED	ANGULAR INTERVAL	No. DATA POINTS
1 rev per 14.4 s	1	360
1 rev per 1.44 s	10	36
1 rev per 0.72 s	20	18
1 rev per 0.351 s	40	9

ASYNCHRONOUS DETECTION:

SPINDLE SPEED	ANGULAR INTERVAL	No. DATA POINTS
14.4 rev/s	1	360

Table 5.1: Spindle speeds for synchronous and asynchronous detection

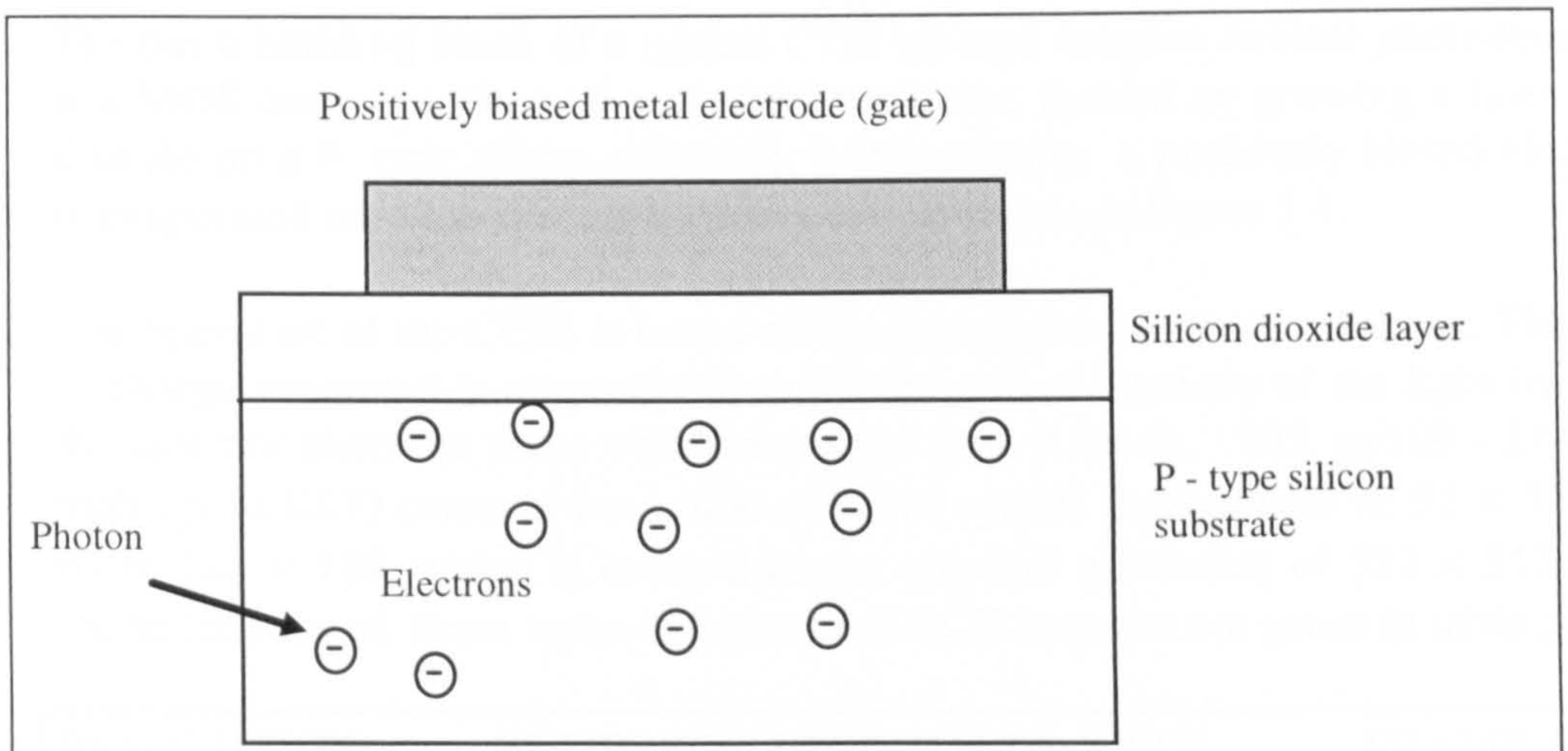


Figure 5.4: Principle of a CCD detector after Gasvik (1995 pp109 -114)

The synchronous and asynchronous rotational speeds of 1 rev per 14.4 sec and 14.4 rev per sec respectively for 360 data points is appropriate for low-speed spindle metrology, ie where the run-out of the spindle typically occurs at 30 - 800 rpm, which is the case investigated in this work.

In this work, 5 batches of 30 data points were sampled synchronously to generate the required error motion polar plot for a spindle speed of approximately 60 rpm.

5.3.3 APPLICATION OF HIGH SPEED CCD ARRAY CAMERAS

5.3.3.1 INTRODUCTION

In section (5.3.2) it was demonstrated that the frame rate of the current (vidicon) camera used in this work cannot adequately detect bearing run-out at spindle speeds greater than 800 rpm. Circumstances where the spindle error motions occur at relatively high speeds (typically 2000 - 3000 rpm) require high frame-rate / non-standard spatial resolution CCD cameras. In this section, a brief description of the application of high frame rate CCD cameras for the measurement of spindle run-out is given.

Currently, high speed cameras with frame rates of up to 10,000 frames/s are available and can be adequate for detecting interferograms of spindles revolving at very high speeds. These type of CCD cameras have non-standard spatial resolution and frame rates, and are commercially availiable*.

5.3.3.2 CHARACTERISTICS OF HIGH SPEED CCD CAMERAS

The basic building block of a typical CCD (charge coupled device) photodetector array, is a MOS (metal oxide semiconductor) capacitor, formed by growing a layer of silicon dioxide on a P- type silicon substrate. It incorporates a positively biased electrode that is evaporated onto the silicon dioxide layer, as shown in figure 5.4.

The operation of the CCD, is based on the generation of electric charges. The amount of charge generated is proportional to the integral of intensity of the light incident on the detector elements taken over the storage time (Gasvik, 1995, pp109 - 114). Some high speed CCD cameras use a non-standard spatial arrangement of 32 × 32 or 64 × 64 or 128 × 128, which is lower than the standard resolution of 512 × 512. Typical characteristics of these types of high speed CCD cameras are given in table 5.2.

RESOLUTION	PHOTOELEMENT DIMENSIONS	DATA-RATE	FRAME-RATE
32 × 32	16 μm × 16 μm	16 MHz	10,000 frames/s
64 × 64	16 μm × 16 μm	16 MHz	2,900 frames/s
128 × 128	16 μm × 16 μm	16 MHz	830 frames/s
256× 256	16 μm × 16 μm	16 MHz	220 frames/s

Table 5.2: Resolution and frame rate of non standard CCD detectors

* Dalsa Inc, 605 McMurray Road Waterloo, Ontario, N2V 2E9, Canada

A typical application of a high speed CCD camera (ie in spindle metrology) for an aerostatic bearing, for example operating at a maximum rotational speed of 3000 rpm (50 Hz), and requiring synchronous run-out measurements at 10 degree angular intervals will require a frame sampling time of 555.555556 μs, since:

ANGULAR MEASUREMENT OF SPINDLE		TIME / FRAME-RATE
10	-	0.555 ms (≅1800 frames/s)
20	-	1.111 ms (≅ 900 frames/s)
30	-	2.222 ms (≅ 300 frames/s)
. . .	.	. . .
. . .	.	. . .
. . .	.	. . .
350	-	0.0194 s (≅ 51 frames/s)
360	-	0.0200 s (= 50 frames/s)

Table 5.3: Frame rate estimation for a bearing rotating at 3000 rpm

In this particular example a high frame-rate CCD camera operating at 2,900 frames/s, with a spatial resolution of 64 × 64 photoelements will suffice, (since the minimum frame-rate required for sampling the interferograms at 10 degree intervals is 1800 frames/s (30 Hz) as shown in table 5.3.

5.4 VIDEO FRAME GRABBER CARD

A video frame grabber board manufactured by Delta Cell* was used to capture, digitise and analyse the interferograms produced by the interferometer. In addition to this an image processing package called Visilog (manufactured by Noesis**) was used to analyse and process the captured interferograms.

---

* Data Cell Limited, Hattori House, Vanwall Business Park, Maidenhead, Berkshire SL6 4UB, UK

** Noesis, Immeuble Nungesser, 13 Avenue Morane Saulnier, 78140 Velizy, France

Features of the frame-grabber card used in this project are:

SPATIAL RESOLUTION:	512 × 512
IMAGE INPUT FORMAT:	Monochrome
IMAGE (FRAME) MEMORY:	512 × 512 × 8 (bits)
FRAME-RATE:	30 Frames/sec
ON-BOARD SPECIAL FEATURES:	External trigger / colour look-up table (LUT)

Table 5.4: Characteristics of frame-grabber

5.5 INTERFEROMETER ADJUSTMENT SERVO-MECHANISM

5.5.1 DESCRIPTION OF SERVO

The interferometric measurement system uses a servo-mechanism to automatically adjust the tilt of the interferometer axis with respect to the axis of rotation of the spindle (under investigation). A detailed description of the interferometer adjustment strategy is given in section 7.2. This section concentrates on the construction of the interferometer adjustment servo.

The interferometer is kenematically attached to the research turning machine tool incorporating the aerostatic spindle, through three legs incorporating adjustment mount screws. Only two legs of the interferometer require servo adjustment. The servos are used to adjust the respective orthogonal axes of the interferometer, and each consist of a DC motor, a gearbox, a pulley and lead-screw as shown in figure 5.5.

Figure 5.5: Photograph showing one of the servo legs

5.5.2 THEORETICAL ANALYSIS OF THE TORQUE REQUIREMENT

The interferometer is positioned by three adjustment lead-screws, and with the aid of figures 5.6 and 5.7 the torque required to raise a third of the weight of the interferometer, is:

$$\frac{3F}{W} = \tan(\alpha + \phi) \qquad (5.3)$$

where, F : applied force
W : the weight of the interferometer
α: angle the reaction force makes with the thread
ϕ : helix angle of the lead-screw thread

Hence

$$F = {}^{W}\!/_{3} \tan(\alpha + \phi) = {}^{W}\!/_{3}\left[\frac{\tan\alpha + \tan\phi}{1 - \tan\alpha \tan\phi}\right] \qquad (5.4)$$

where, $\tan\alpha = \frac{P}{\pi d}$

$\tan\phi = \mu$

torque $T = \frac{Fd}{2}$

where, P: thread pitch

d: thread diameter

Equation (5.4) can be modified to give:

$$\text{Torque } T = \frac{W.d}{6}\left[\frac{\left(P/\pi d\right)+\mu}{1-\mu.\left(\frac{P}{\pi d}\right)}\right] \qquad (5.5)$$

where, μ: coefficient of friction

The interferometer alignment system has the following variables:

P = 0.6 mm

d = 12 mm

W = 500 N

μ = 0.22

Substituting the above into equation (5.5), the theoretical value of the torque, is approximately 0.3 Nm.

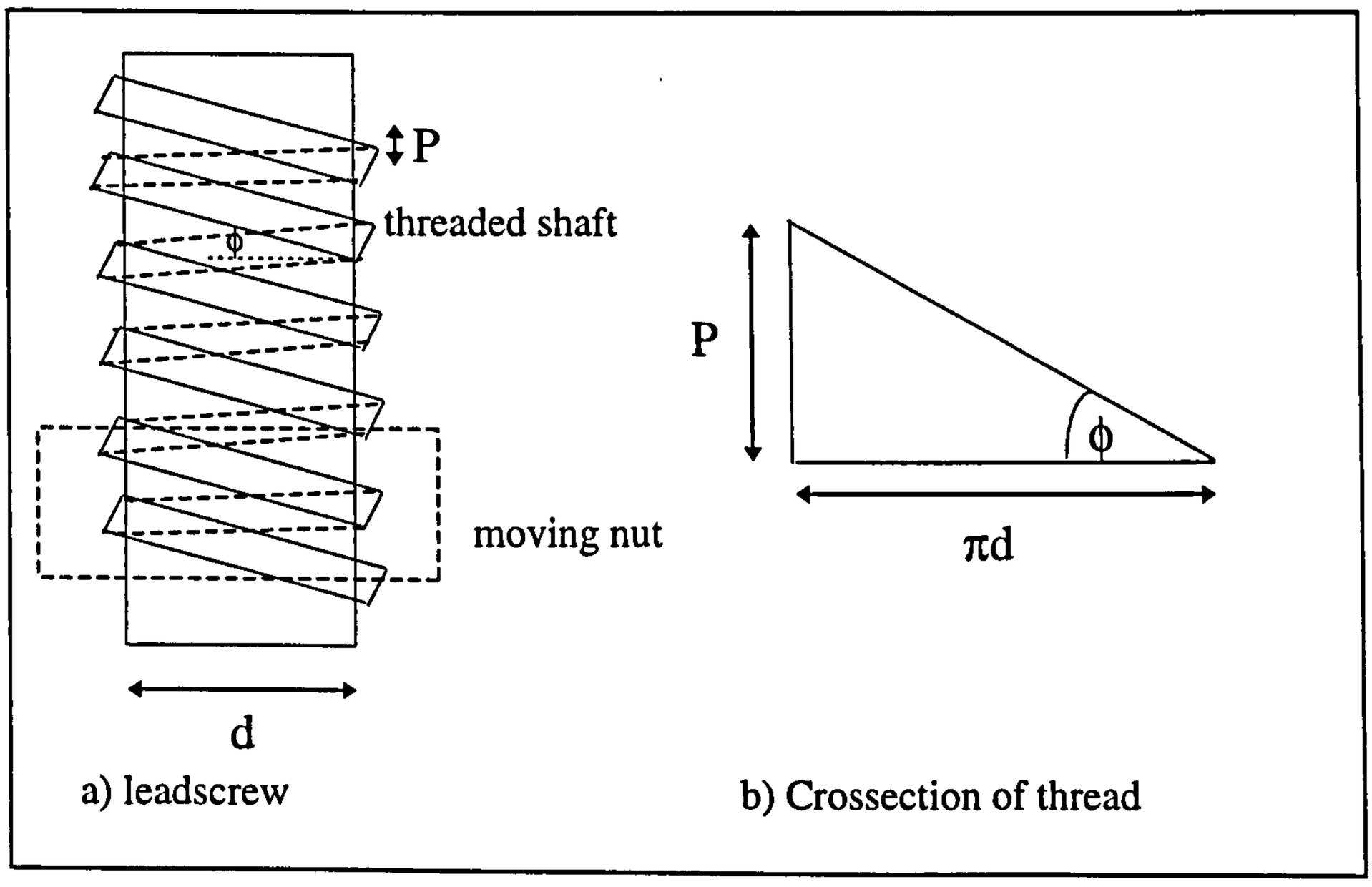


Figure 5.6 : Schematic of the forces that act on a section of a lead-screw

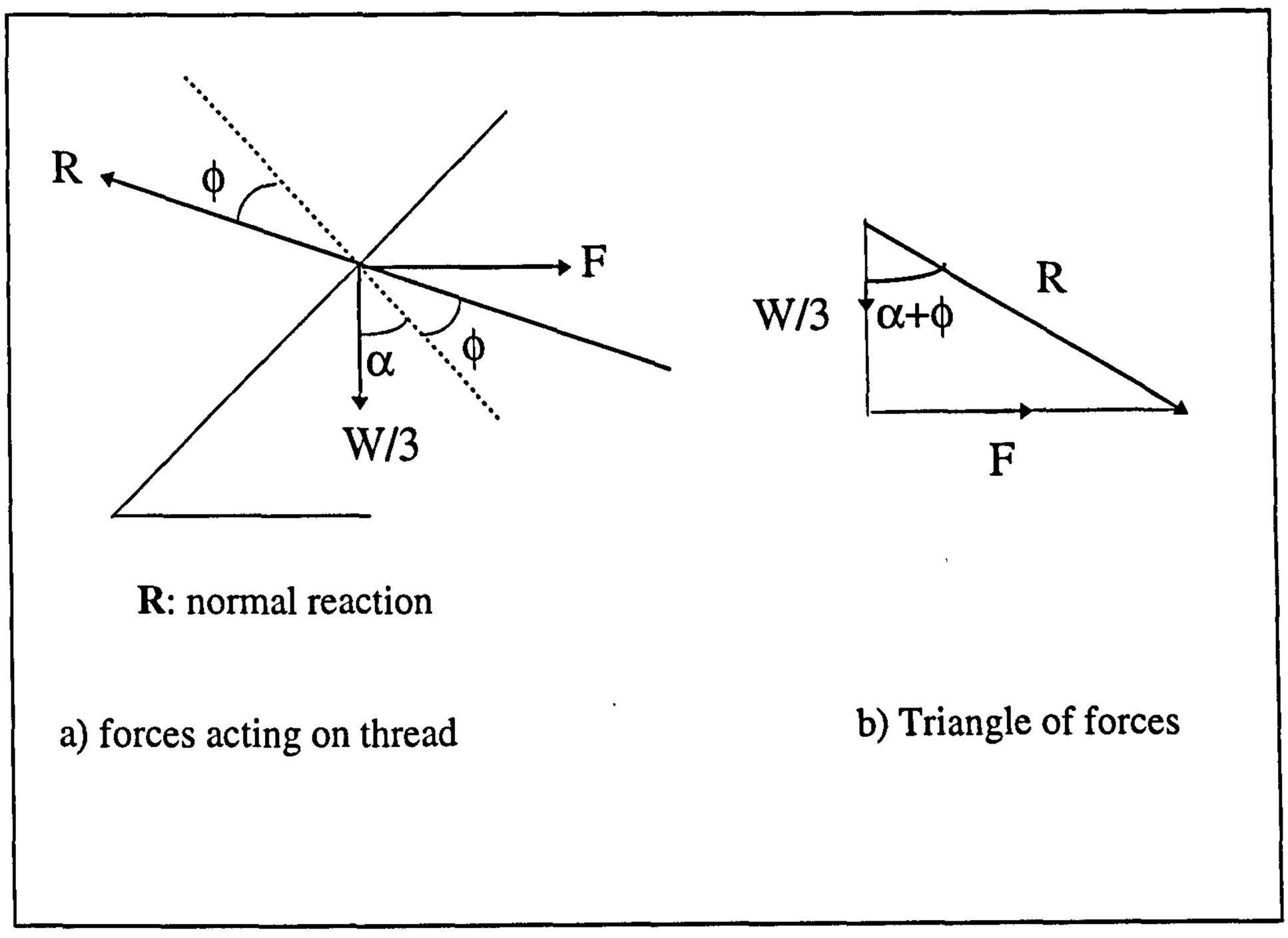


Figure 5.7: Free body diagram for the section of the lead-screw

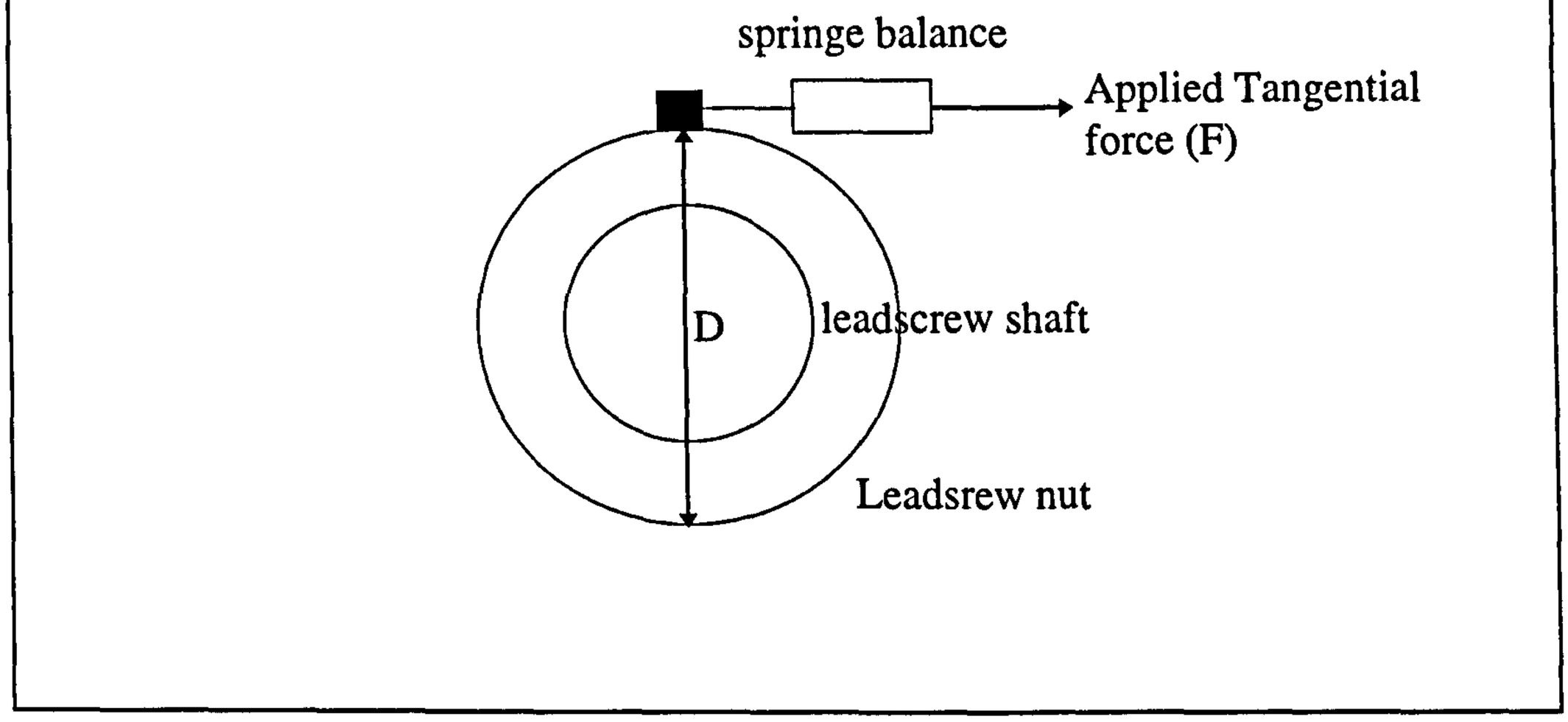


Figure 5.8: Plan view schematic of direct measurement of servo torque

5.5.1.2 EXPERIMENTAL ANALYSIS OF THE TORQUE REQUIREMENT

Analysis of the interferometer adjustment (servo mechanism) is based on measuring the applied tangential force F required to turn the adjustment nut using a spring balance (as shown schematically in figure 5.8), and then multiplying this value by the outer diameter of the nut (D).

the applied torque value (T) is: FD (5.6)

where, F = 12N and D = 40 mm

$T = 12 \times 40 / 1000 \approx 0.5$ Nm

5.5.1.3 SPECIFICATION OF THE DC MOTOR / GEAR BOX

The value of chosen torque for the DC motor/gearbox is 20 multiplied by the empirical torque value, where 20 is a safety factor (which takes in to account the high torque requirement at very low speeds). This gives a torque requirement of:

$0.5 \times 20 = 10$ Nm.

The servo mechanism used in this project incorporated a *Precision Portescap* DC motor and a *Synchronous* gearbox both supplied by RS*, to provide the required torque drive.

Specifications of the selected DC motor / gearbox are given tables 5.7 and 5.8.

PORTESCAP PRECISION DC MOTOR SPECIFICATION	
Supply voltage:	12 V dc
Motor torque:	8.7 mNm
Power-output:	3.8 W
Speed output:	5800 RPM

Table 5.6: Features of servo motor

* RS Components, PO Box 99, Corby, Northants NN17 9RS

SYNCHRONOUS GEARBOX SPECIFICATION	
Gearing ratio:	3600:1
speed output:	5800/3600 = 1.6 RPM
Torque output:	(8.7×3600) / 1000 = 31.32 Nm

Table 5.7: Features of servo gearbox

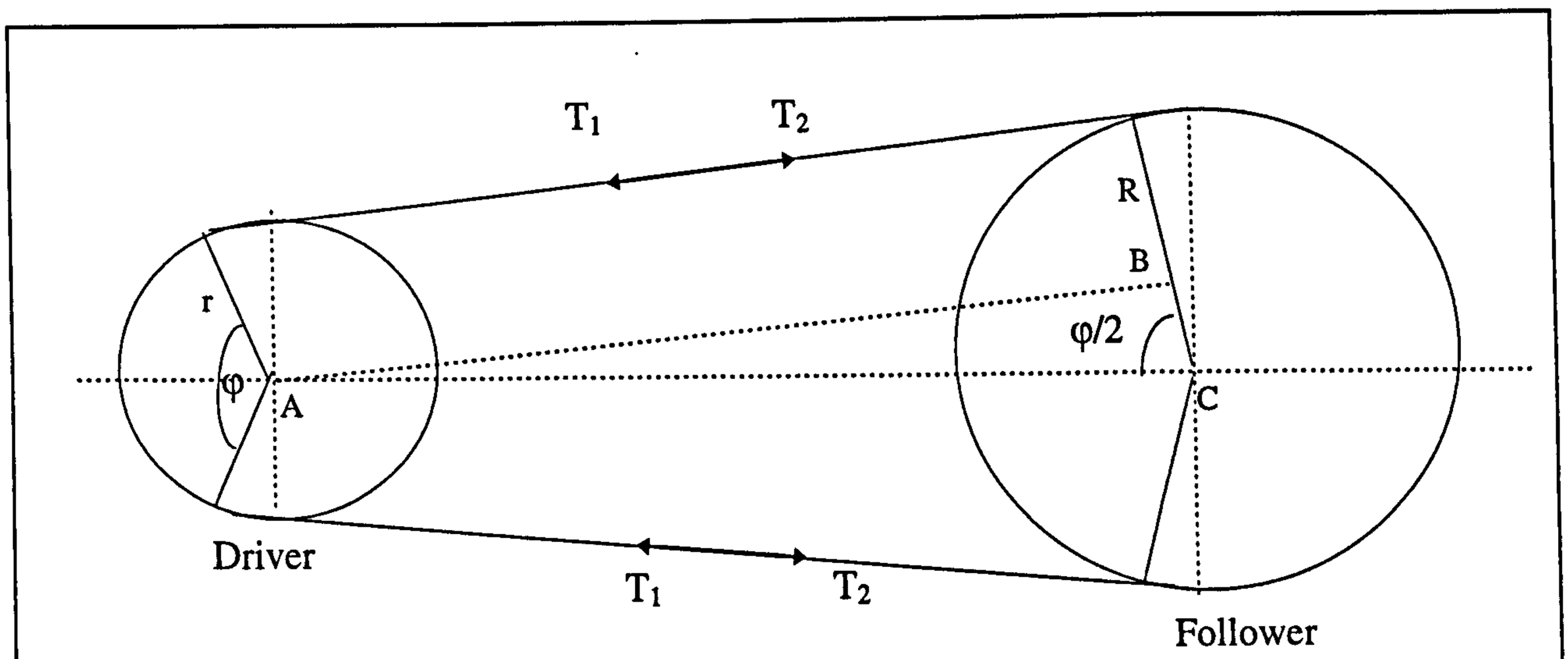


Figure 5.9 : Plan view of pulley mechanism

As shown in table 5.8, the torque output of 31.32 Nm used for the servo mechanism is three times greater than the experimental torque value of 10.0 Nm (estimated in the previous page). This is due to the low speed and high torque required by the servo mechanism for the precise (tilt) control of the interferometer.

5.5.1.4 PULLEY MECHANISM

A pulley mechanism is used in each servo, for transmitting the required torque to the interferometer adjustment screws. Each pulley consists of two wheels and a toothed belt. Figures 5.9 and 5.10 show the plan and side view of the pulley mechanism used to drive one of the interferometer adjustment arms. Ideally the mechanism is only required to transmit the torque from the DC gearbox/motor to the interferometer screw without any further gearing. However, in order to further compensate for torque, the diameter of the follower wheel is one and a half times that of the driver wheel.

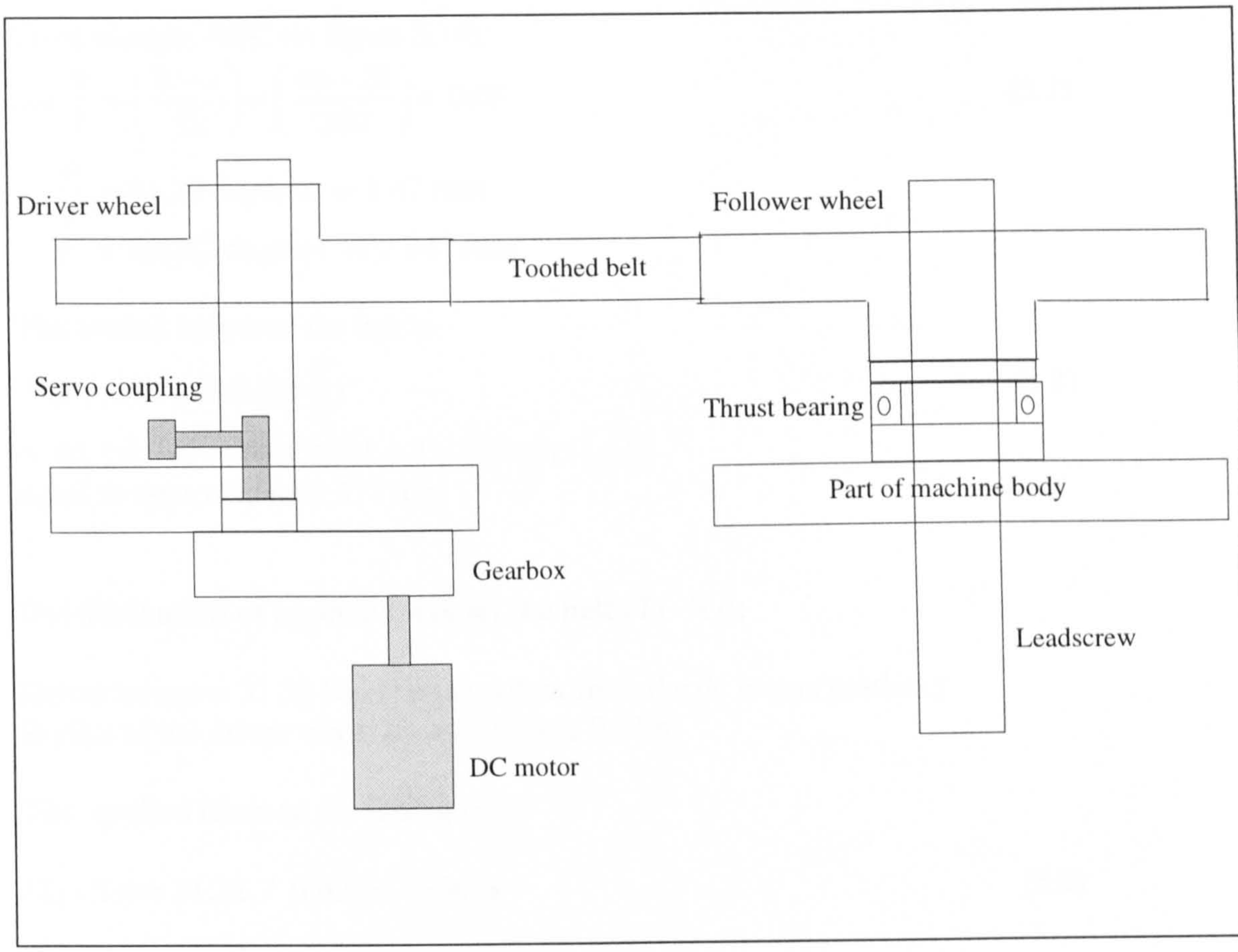


Figure 5.10 : Side view of pulley mechanism

5.5.1.5 SPECIFICATION OF TOOTHED BELT PULLEY

The timing belt pulley was supplied by HPC*, specifications for the belt mechanism are as follows:

Belt Pitch : 10 mm
Belt width : 10 mm
Outer radius (follower) : R = 40 mm
Outer radius (driver) : r = 20 mm
Distance between pulley : L = 200 mm

Table 5.8: Dimensions of servo pulley

---

* Hinchliffe Precision Components, Storforth Lane Trading Estate, Chesterfield S41 0QZ

Determination of the required length of the belt based on the design shown in figure 5.9:

From triangle ABC (in figure 5.14),

$$\cos\frac{\varphi}{2} = \left(\frac{R-r}{L}\right) = \left(\frac{40-20}{200}\right) = 0.05 \qquad (5.7)$$

$\frac{\varphi}{2}$ = 84.26 degrees or 1.47 rads

φ = 168.52 degrees or 2.941 rads

The overall length of the belt is:

$$R\varphi + r\varphi + 2(L \sin\frac{\varphi}{2}) \qquad (5.8)$$

$= 40 \times 2.941 + 20 \times 2.941 + 2 \times 200 \sin(1.47)$

equal to approximately 573 mm

Determination of applied force on the belt (T_1 - T_2):

Driver torque = 31.32 Nm (torque output from the dc motor/gearbox)
Radius of the driver wheel (r) = 20 mm or 0.02 m

The applied force on the timing belt:

$(T_1 - T_2) = 31.32 \ / \ 0.02 \ = \ 1556$ N (5.9)

Estimation of the torque applied to the follower after transmission:

Follower torque = $(T_1 - T_2)0.04 = 62.24$ Nm . (5.10)

Validation of the chosen belt based on its power transmission capacity:

The power transmission capacity of the chosen belt is approximately 0.5 kW (according HPC). The power input of the driver wheel of the interferometer adjustment system is given by :

$(T_1 - T_2) \times$ speed (5.11)

where, $(T_1 - T_2)$ = 1556 N (ie calculated above)

speed = 1.6 (ie given in table 5.6)

Hence the power transmitted by the pulley belt is: $1556 \times \frac{2\pi}{60} \times 1.6 = 0.261$ kW

Since this is less than the power transmission capacity of the belt (ie 0.5 kW), it is within design limitations.

5.5.1.6 MISCELLANEOUS COMPONENTS OF SERVO

Each servo of the interferometer incorporates the following ancillary components:

Driver shaft: is used to transmit the torque from the gearbox/motor to the pulley driver wheel.

Coupling component: the gearbox shaft and driver (wheel) shaft are coupled together using a *grub-screw*.

Thrust bearing washer is used to reduce the friction in the movement of the interferometer lead-screw driven by the pulley follower wheel (shown in figure 5.10).

5.5.2 ASSEMBLY OF SERVO COMPONENTS

Figure 5.11 shows the side view of the interferometer, the machine structure and one of the mounted servo-mechanisms. In order for the interferometer to accommodate the auto-adjustment (servo-mechanism), two plates were required on the surface of the machine structure. Both plates were made from aluminium and have geometric dimensions of 70 mm by 150 mm, it is anticipated that there should be no serious effects of vibration, or any other form of mechanical failure, as long as the gearbox / motor arrangement are accurately assembled to the respective plates.

Figure 5.12 shows a schematic of one of the plates, indicating a central slot used for adjusting the driver (wheel) shaft, and two holes used for fastened the plate on to the machine structure.

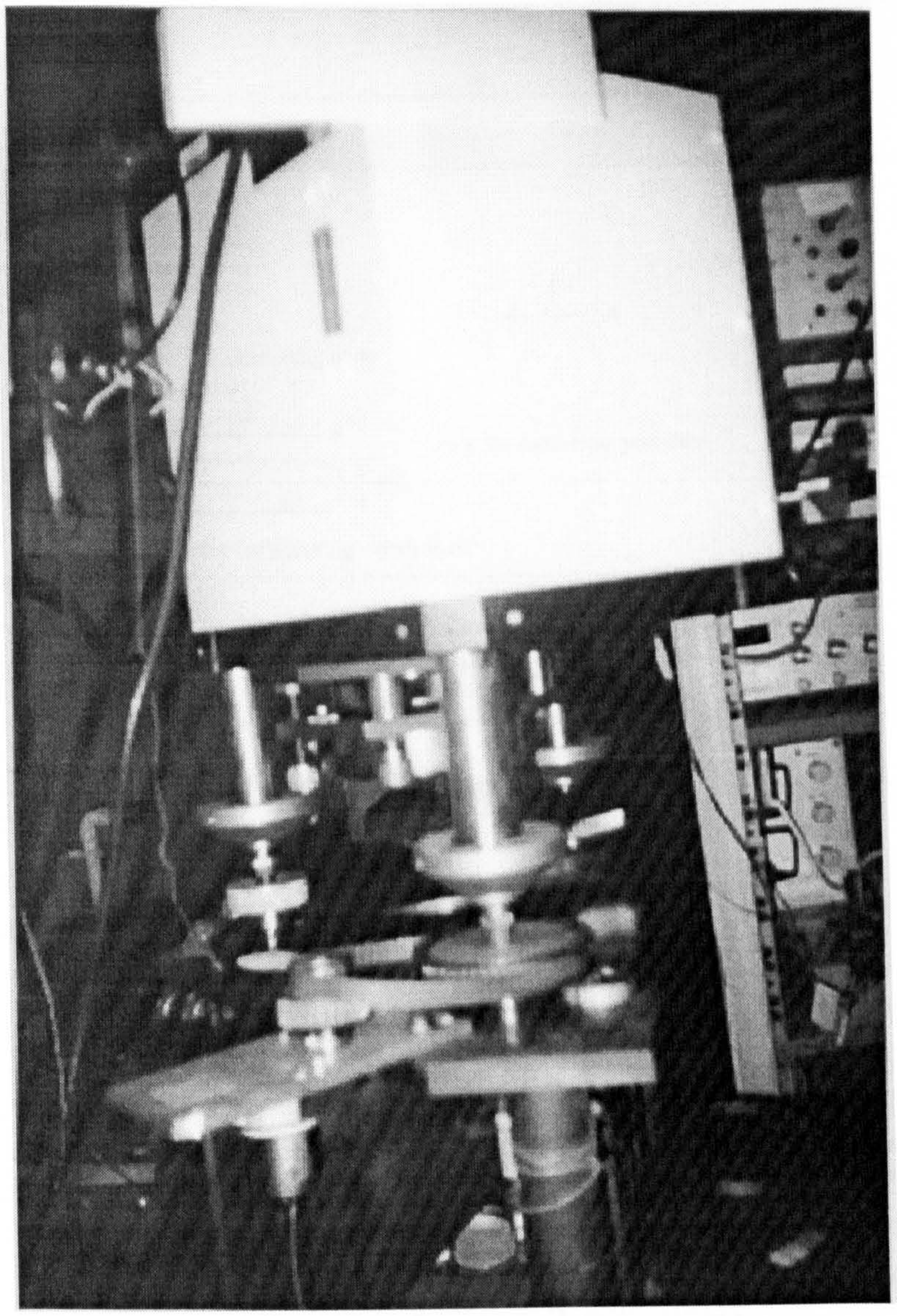


Figure 5.11: Side view (photograph) of Fizeau interferomerter, its adjustment servo and machine incorporating the air spindle

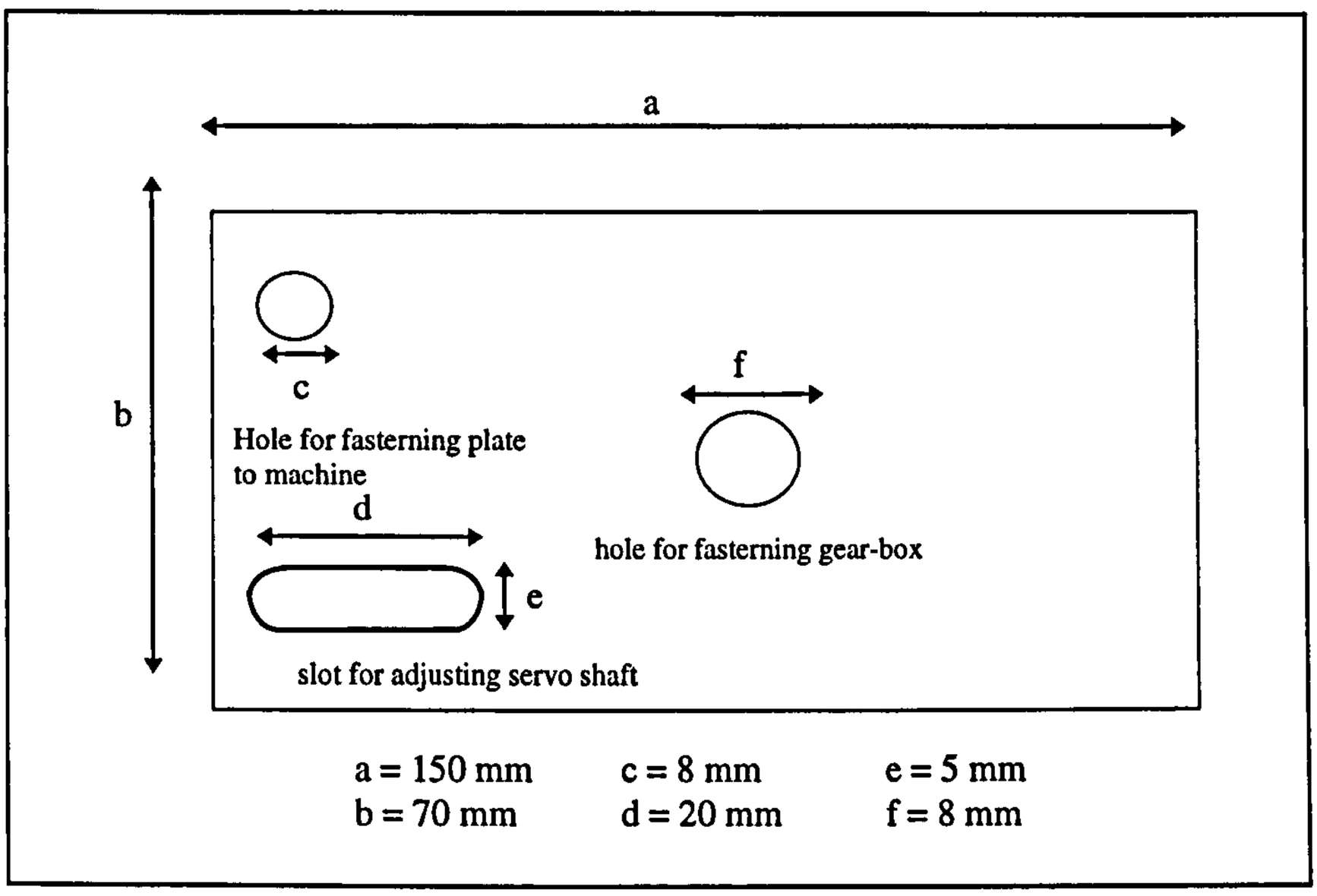


Figure 5.12: Schematic of plate

5.5.3 SERVO-MECHANISM ELECTRONICS

The servo-mechanism incorporates manual / computer control and dither switching circuits. The function of the control switching circuit is twofold, ie it is used to used to select a particular servo adjustment screws (X or Y) and its direction of motion (clockwise or anticlockwise). The dither switching circuit on the other hand is used to *sweep* the operating point of the servo output over a certain range around its nominal position, on a time-scale much shorter than that of its dynamics in order to reduce non-linearity problems such as backlash (Cook, 1994).

Components for the control/dither switching circuit used for controlling a single adjustment arm of the servo are listed in table 5.9.

12 V DC power supply unit
12 V AC (10 Hz) dither supply unit
NPN transistors
single pole change-over (SPCO) relays
double pole change-over (DPCO) relays
inverters
35 K resistors
1N4000 diodes
L.E.Ds to indicate the direction of the servo

Table 5.9: Components of control/dither switching circuit

Figure 5.13 shows the circuit diagram for the switching electronics required for controlling the clockwise and counter-clockwise motion of the servo-mechanism dc motor.

5.5.3.1 COMPUTER CONTROL SWITCHING ELECTRONIC CIRCUIT

As shown in figure 5.13, two output signals or bits are required to control the status of the DC motor. Depending upon what signal is sent to the two output pins, the motor is required to have the following status:

BIT#1	BIT#2	STATUS
0	0	SERVO OFF
1	0	CLOCKWISE MOTION OF SERVO
1	1	ANTI-CLOCKWISE MOTION OF SERVO
0	1	SERVO OFF

Table 5.10: Status of control/ dither digital bits

Other control signals used are:

BIT	STATUS
1	DITHER CLOCKWISE
0	DITHER ANTICLOCKWISE
1	SAFETY BIT ON
0	SAFETY BIT OFF

Table 5.11: Ancillary status of control/dither bits

Configuration of servo (on/off), servo (clockwise/anticlockwise), dither (clockwise/anticlockwise) and safety (on/off) logic inputs must be programmed in the appropriate manner to achieve the required mode of operation of the servo. The servo has three modes of operation, these are: *AC -DC mode* (ie when the servo DC supply signal is dithered with the AC signal), *AC mode* (ie when the servo input is the AC dither supply signal) and *DC mode* (ie when the servo input is the DC supply signal). Figure 5.13 shows a the circuit diagram of the AC-DC mode of operation.

Selection of these modes of operation require the following inputs:

SERVO MODE (AC-DC):	
INPUT	STATUS
DITHER POWER SUPPLY	1
SERVO ON/OFF	1
SERVO DIRECTION	1 or 0
DITHER DIRECTION	1 or 0
SAFETY BIT ON/OFF	0

Table 5.12: Status of servo mode (AC-DC)

SERVO MODE (AC):	
INPUT	STATUS
DITHER POWER SUPPLY	1
SERVO ON/OFF	0
SERVO DIRECTION	1 or 0
DITHER DIRECTION	1 or 0
SAFETY BIT ON/OFF	1

Table 5.13: Status of servo mode (AC)

SERVO MODE (DC):	
INPUT	STATUS
DITHER POWER SUPPLY	0
SERVO ON/OFF	1
SERVO DIRECTION	1 or 0
DITHER DIRECTION	1 or 0
SAFETY BIT ON/OFF	0

Table 5.14: Status of servo mode (DC)

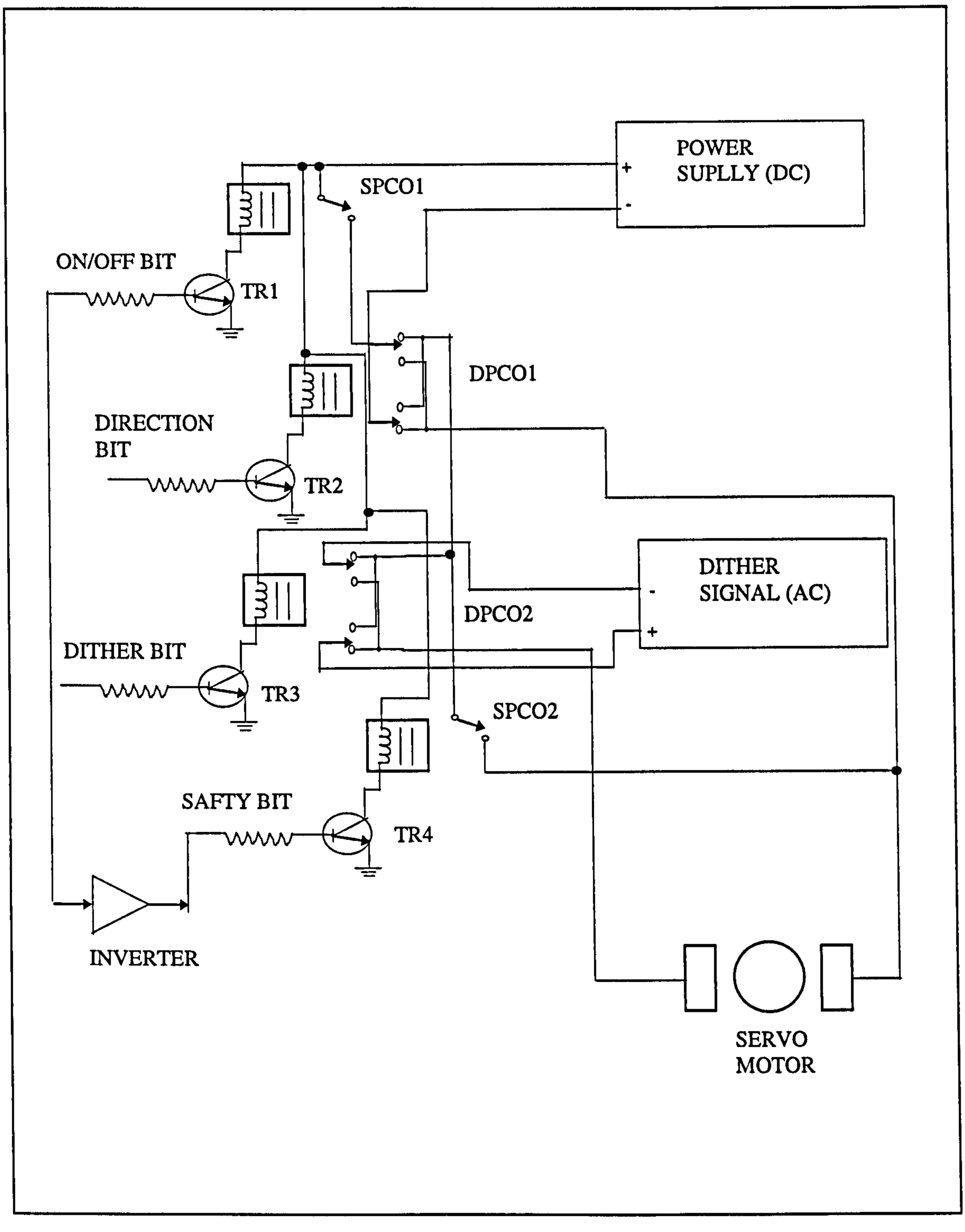


Figure 5.13: Computer control / dither circuit diagram for one of the servo arms

In figure 5.13, transistor TR1 is used to switch on the single-pole change-over relay SPCO1 when a small current I_1 is applied (ie HIGH state) to the base connection of the transistor. Similarly when a small current I_2 is also applied to base connection of transistor TR2, an amplified collector current I_c is used to energise the double pole change-over relay DPCO1 to change the polarity of the 12 V power supply. Switch SW1 is used to select the appropriate direction of the dither signal by energising relay DPCO2. The safety bit labelled TR4 avoids any conflicts in signals between the dither and DC servo supply inputs. An inverter gate is used to invert the input of the servo on/off bit, output of the inverter gate is used as the input of safety (on/off) bit.

Protection diodes are used to suppress any high voltage spikes which might be produced across the coils of the respective relays (ie not shown in the figure). A resistor is used to limit the current passed on to the respective transistors to avoid damage. Calculation of the required value (R_b) of this resistor is given as follows:

$$I_C = \beta I_b \qquad (5.12)$$

$$V_{TTL} = I_b R_b + V_{BE} \qquad (5.13)$$

$$I_C R_L + V_{CE} = V_S \qquad (5.14)$$

where, β: transistor gain
R_L: the relay (inductor) resistance = 960 ohms
V_{CE} : the collector-emitter voltage at saturation = 0.2 V
V_S : the supply voltage = 12 V
R_b: the resistance of the unknown resistor
V_{BE}: the base-emitter voltage at saturation = 0.7
V_{TTL}: = the TTL voltage signal from the PC interface card = 5 V

from equation (5.13) :

$$I_C = (V_S - V_{CE}) / R_L \qquad (5.15)$$

$$= 0.0123 \text{ A}$$

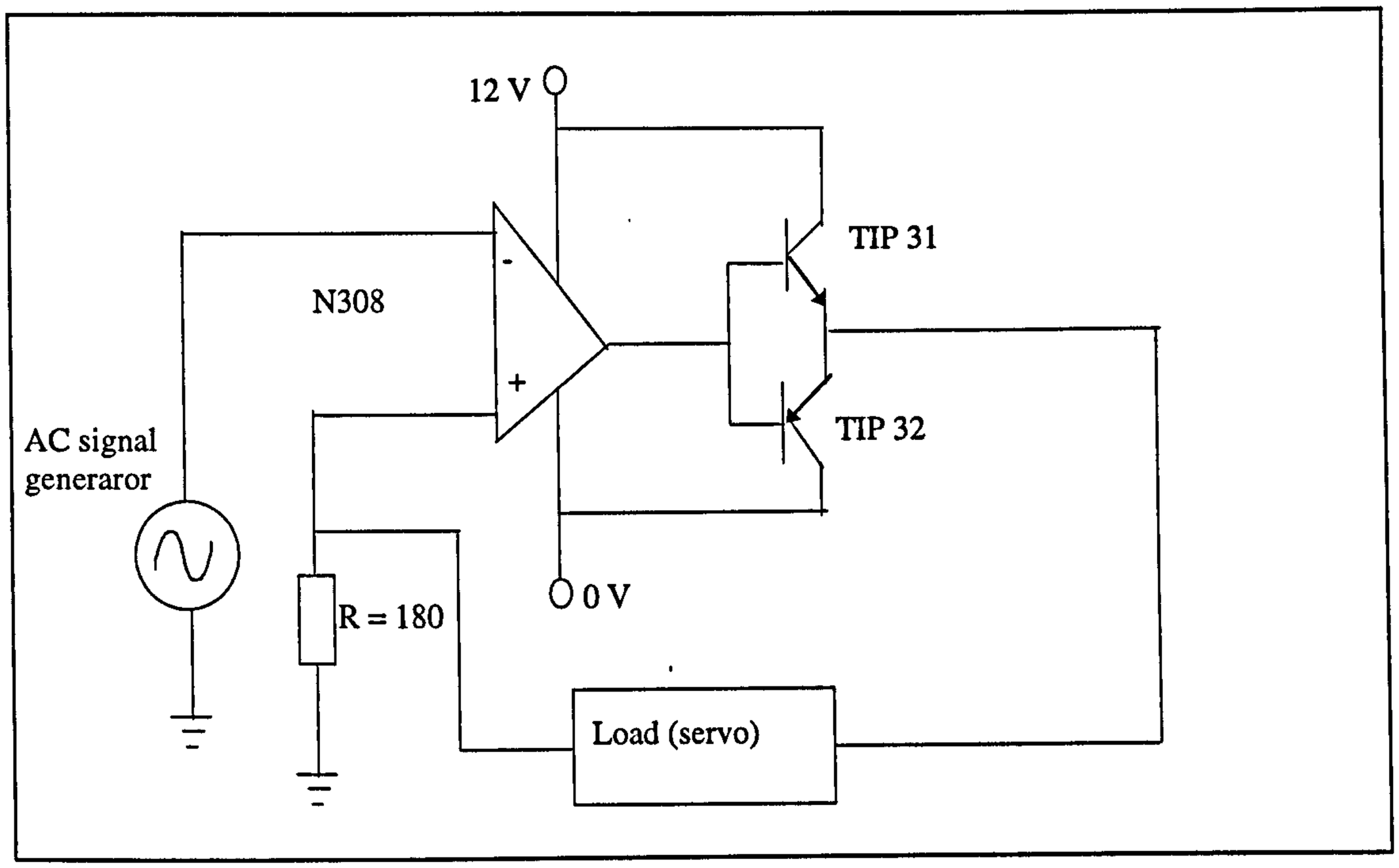


Figure 5.14: Dither signal circuit diagram

from equation (5.12) :

$I_b = I_C / \beta = 123\ \mu A$

from equation (5.3)

$R_b = (V_{TTL} - V_{BE}) / I_b = 35$ K ohm (to 2 decimal places)

So a 35 K ohm resistor would be required to limit the current passing through the transistor.

5.5.3.1 AC DITHER CIRCUIT

The output of the servo mechanism is continuously *dithered* over a range around its nominal position in order to reduce backlash (see section 7.72 for a detailed explanation of servo backlash reduction). Figure 5.14 shows a schematic of the dither circuit, the dither (sinusoidal) signal is generated with an AC signal generator and amplified with the aid of an N308 op-amp and two TIP (high power) current amplifier transistors.

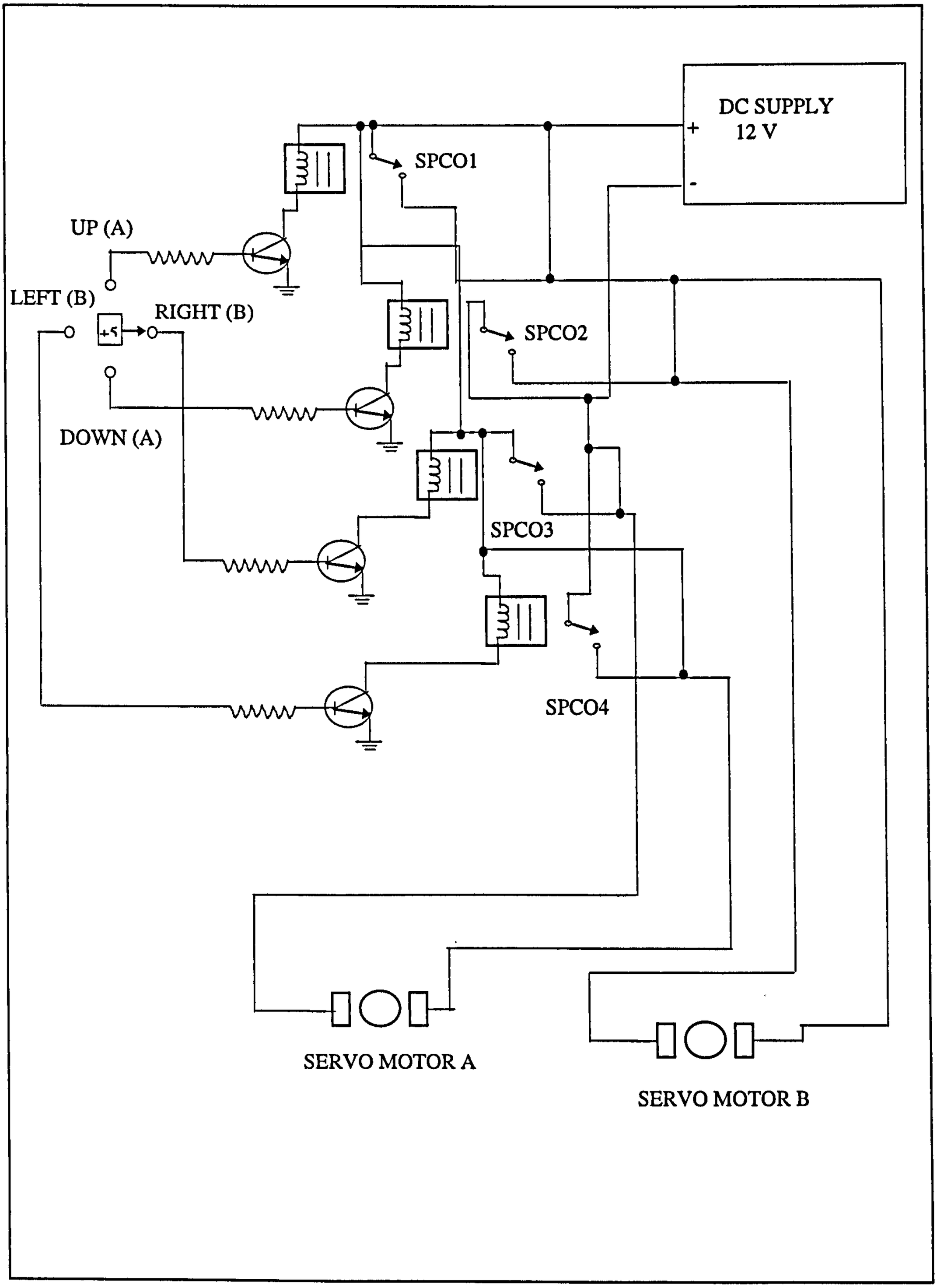


Figure 5.15 : Switching circuit for the interferometer servo - manual control

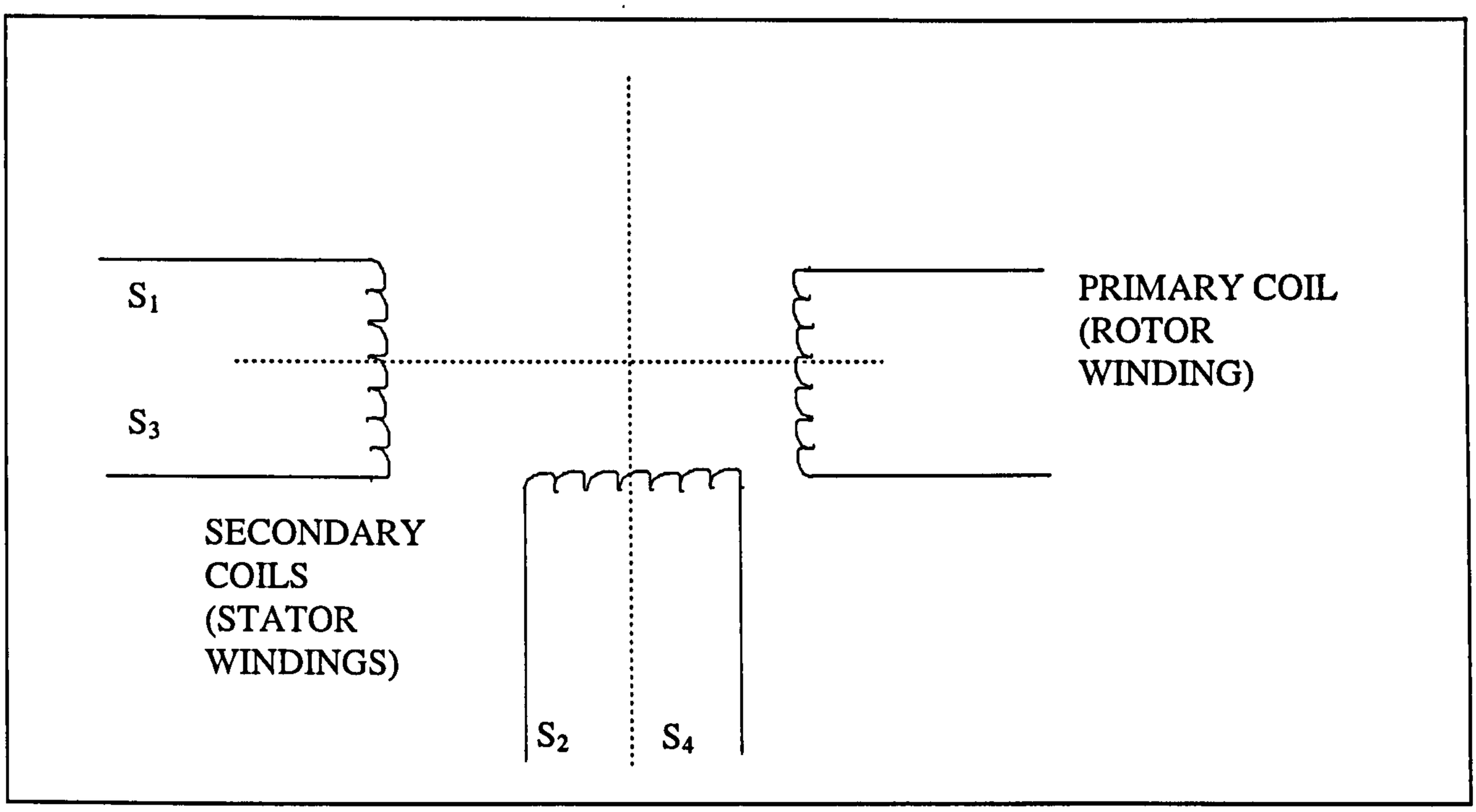


Figure 5.16 : Schematic of a resolver

5.5.3.2 MANUAL CONTROL SWITCHING ELECTRONIC CIRCUIT

In addition to automatic (computer) control, the servo mechanism incorporates manual control for normal type adjustment of the interferometer. Figure 5.15 shows the circuit diagram of the servo manual control electronics. The figure depicts the switching of the servo with the aid of a joystick, for controlling both the X and Y interferometer screws. Configuration of the joystick control is:

JOYSTICK MOTION	DC MOTOR	MOTOR DIRECTION
UP	X	Clockwise
DOWN	X	Anti-clockwise
RIGHT	Y	Clockwise
LEFT	Y	Anti-clockwise

Table 5.15: Status of servo manual (joystick) control

The servo control is switched from *computer* to *manual* control with the aid of a double-pole-double-toggle switching circuit .

5.6 RESOLVER AND ITS DIGITAL CONVERTER

5.6.1 OPERATION OF THE RESOLVER

The precision spindle used in this work is equipped with a resolver for measuring the angular position of the rotating spindle rotor. Operation of the resolver requires the application of an AC carrier (reference) signal at the rotor terminals (primary coil) R_1

and R_2, which then produces a pair of orthogonally-phased modulated signals at the isolated stator windings (secondary coils) S_{24} and S_{13} as shown schematically in figure 5.16. The voltage modulated signals induced in the stator windings are:

$$S_{24} = KE\sin\omega t\cos\theta \quad (5.16)$$

$$S_{13} = KE\sin\omega t\sin\theta \quad (5.17)$$

where K : is the transformation ratio
θ : is the shaft (spindle rotor) angular position from some reference point
ω : is the supply $2\pi f$ carrier frequency
E : supply voltage

Essentially a resolver has similar characteristics to a variable transformer (Slocum 1992, pp 147-154), where the output depends on the angle the secondary coil makes with respect to the primary. The magnitude of the electromagnetic coupling between the primary and the secondary coils of the resolver determine the output voltages of the stator windings, whose amplitude is proportional respectively to the sine and the cosine of the angular position of the rotor. Measurement of angular position of the spindle (rotor) is directly due to the modulation of the reference (carrier) signal as the spindle revolves.

5.6.2 INTERFACING THE RESOLVER TO THE COMPUTER

5.6.2.1 INTRODUCTION

In order to interface analogue signals obtained from the reslover to a computer, a device known as a resolver-to-digital converter (RDC) is used. The RDC used in this project has a 16 - bit resolution (which is equivalent to 96 μrad 20 arc seconds), supplied by the DDC company*.

There are many different ways of converting the resolver analogue signal into a digital word using the following RDCs (Slocum, 1992 p151-153):

- Phase shift converters
- Function generator converters
- Successive approximation converters
- Harmonic oscillator converters
- Tracking converters (ie method of conversion used in this project)

* DDC (UK) , Mill Reef House, 9/14 Cheap Street, Newbury Berks RG14 5DD

5.6.2.2 MODE OF OPERATION OF TRACKING RDC

The tracking RDC multiplies signal pairs from the resolver stator with the aid of special sine and cosine digital-to-analogue converters (DACs), as shown in schematically in figure 5.17 (DDC, 1994). This process converts a measured digital angle (ϕ) into sine and cosine wave-forms:

$$V_1 = S_{13} \sin\phi = KE \sin \omega t \sin\theta \sin\phi \qquad (5.18)$$

$$V_2 = S_{24} \cos\phi = KE \sin \omega t \cos\theta \cos\phi \qquad (5.19)$$

The difference between these signals is the *error* signal (which has the same frequency as the carrier):

$$KE \sin \omega t \sin\theta \sin\phi - KE \sin \omega t \cos\theta \cos\phi = KE \sin \omega t \sin(\theta - \phi) \qquad (5.20)$$

According to Slocum (1992, p152) the magnitude of the error signal given in equation (5.19) is amplified, and a phase sensitive detector is used to demodulate the signal and provide a DC voltage proportional to $\sin(\theta - \phi)$. The error signal is fed to an integrator and then to a voltage proportional oscillator (VCO) which drives a digital counter (as shown in figure 5.17). Output from the digital (up/down) counter represents the digital value of the resolver, which is strobbed into latches, and subsequently ready to be read to a digital I/O port.

5.6.2.3 CHARACTERISTICS OF THE CHOSEN RDC

The tracking resolver-to-digital converter used in this project is the monolithic RDC-19220. It incorporates features such as programmable angular resolution (which varies from 10 to 16 bits) and tracking rates (which is a measure of the maximum speed the sensor can acquire angular position data). A summary of the characteristics of the sensor (DDC, 1994) is given in table 5.16.

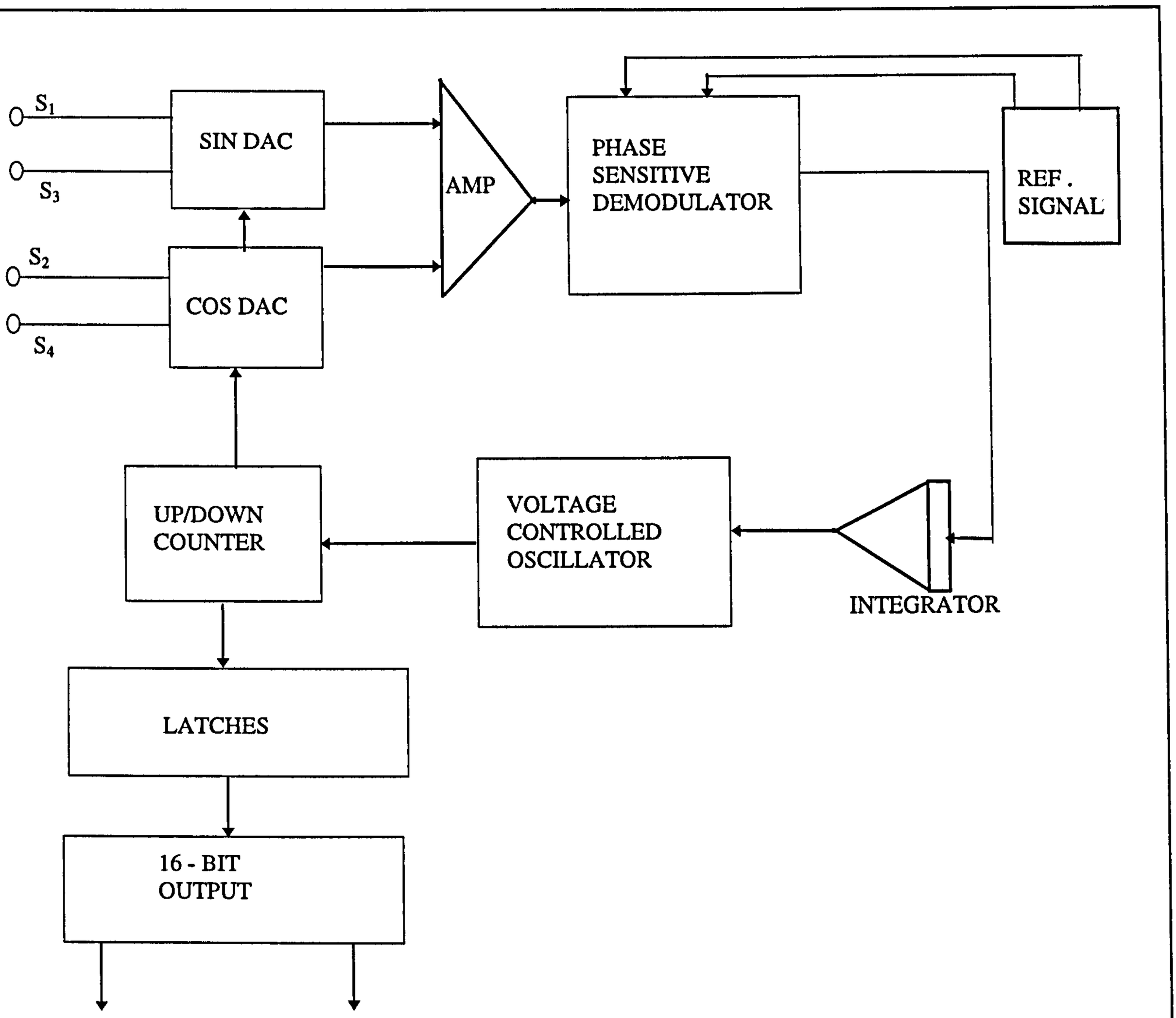


Figure 5.17: Schematic of the tracking RDC

RESOLUTION	10,12,14 or 16 bit resolution and is programmable
INPUTS	V_1 and V_2 inputs are 2 V (rms) Resolver reference input is +7 and -7 V (max) at 40 kHz (max)
INPUT IMPEDANCE	10 M ohms
OUTPUTS	TTL/CMOS inputs with LOW = 0.8 V (max) and HIGH = 2 V (min) and 5 V (max)
ACCURACY	8,4,2,1 least significant bits (depending on the selected resolution)
TRACKING RATE	2300 revs/min (max), and is programmable
BANDWIDTH	1200 Hz (max), and is programmable

Table 5.16: Characteristics of RDC

Figure 5.18 shows a schematic of the interface of the resolver to a personal computer used in this project for data logging, via the RDC. The figure shows passive components such as resistors and capacitors used for attenuating the RDC inputs and for setting the required bandwidth (tracking rate) of the converter.

Factors that must be considered when using the 19220 RDC are:

- impedance matching between the resolver and RDC
- attenuation of the input signals (ie reference, and stator signals)
- improving the bandwidth and tracking of the converter
- screening the signal and power supplies

5.6.2.4 IMPEDANCE MATCHING

One of the features of RDC chip is that it has a relatively high input impedance of 10 MΩ, implying that resolver supply signal (V_s) must have a relatively low impedance. Figure 5.19 shows a *equivalent* circuit of resolver and RDC. The current flowing through the circuit is I_i and the AC supply voltage of the resolver is:

$$V_s = I_i(Z_s + Z_i) \qquad (5.21)$$

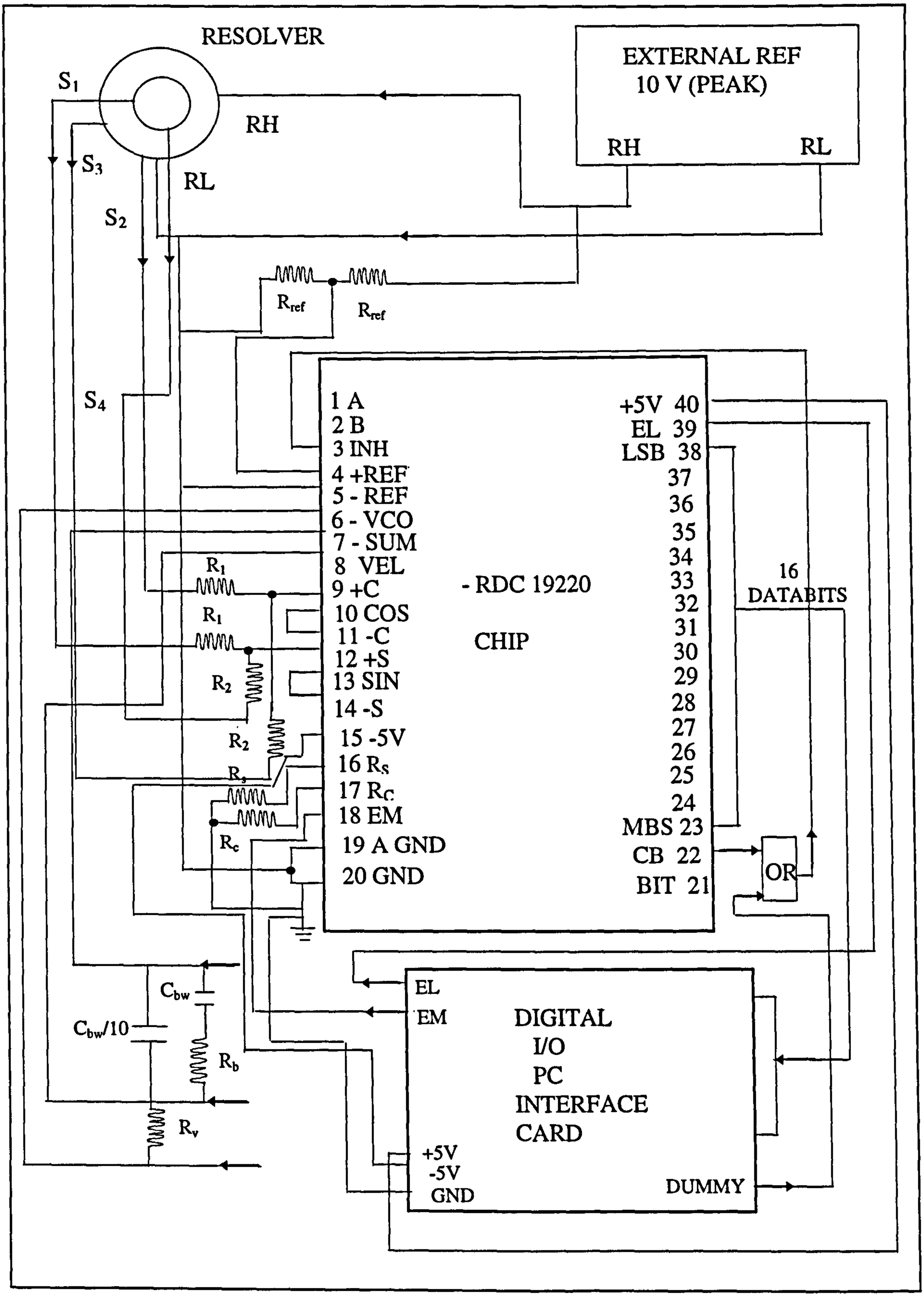


Figure 5.18: Interface of RDC with PC interface card

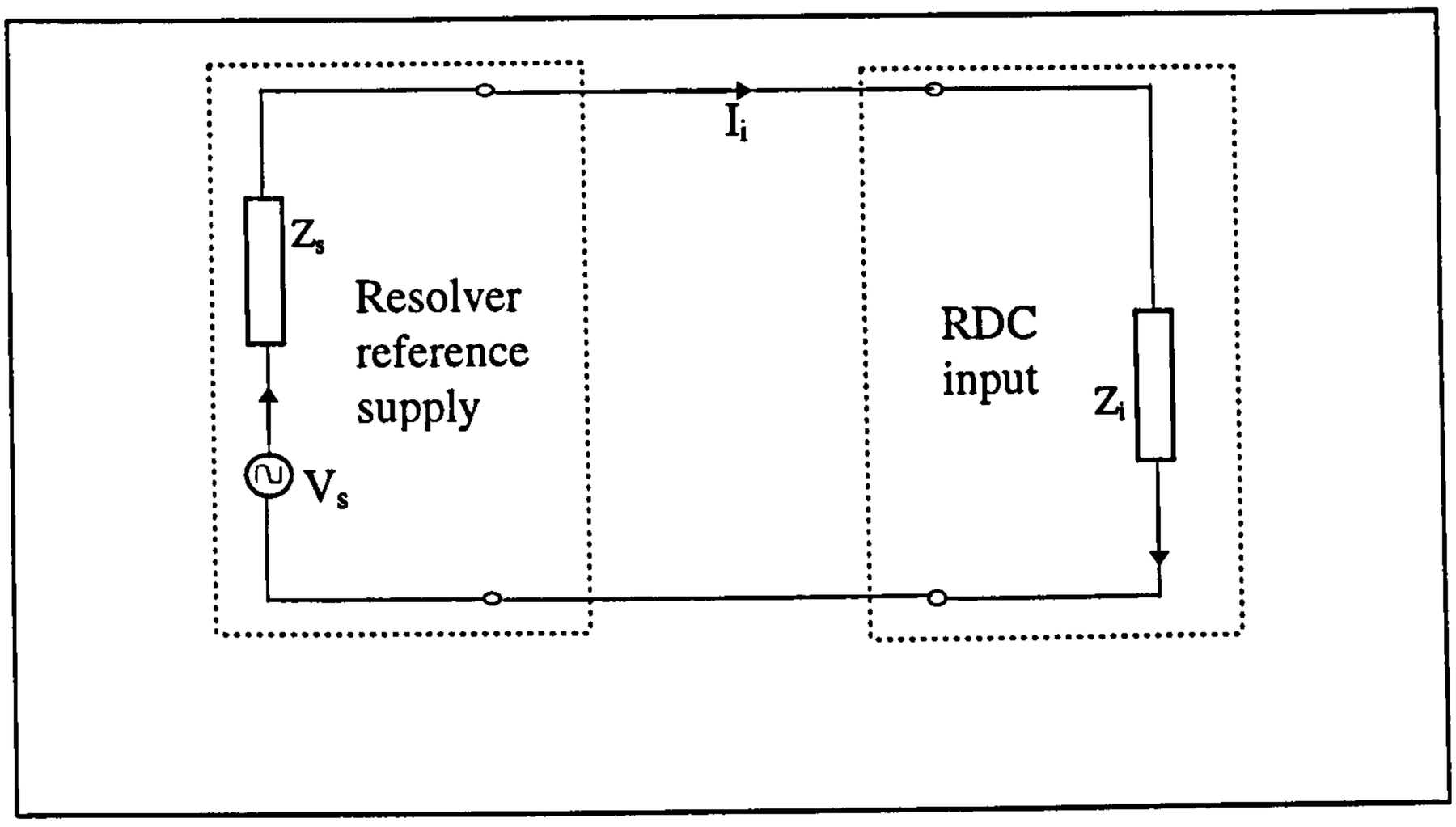


Figure 5.19: Equivalent circuit of for RDC and its input

where Z_i and Z_s are the respective impedance values of the RDC and resolver and

$$V_i = I_i Z_i \tag{5.22}$$

Substituting equation (5.21) into equation (5.22):

$$V_i = V_s\left(\frac{Z_s}{Z_s + Z_i}\right) \tag{5.23}$$

where V_i is the RDC input value in volts.

Typical values of impedance and voltages used in this work are:

$Z_s = 1$ K
$Z_i = 10$ M
$V_s = 10$ V (peak)

substituting these values into equation (5.22) gives $V_i = 9.9099$ V, and the amount of signal attenuation is :

$$\frac{V_s - V_i}{V_s} = 9.901 \times 10^{-3} \tag{5.24}$$

The impedance of the resolver supply and stator inputs is relatively small compared with the input impedance of the RDC (ie note that the stator inputs are derived from the resolver reference supply as shown in figure 5.19), hence a only negligible amount of signal loss occurs.

5.6.2.5 ATTENUATION OF THE INPUT SIGNALS

Attenuation of the input signals is required in order to match the required voltage requirements of the RDC chip. This can be accomplished by forming a potential divider circuit across the inputs, with the aid of resistors (R_1 and R_2) as shown in figure 5.18.

5.6.2.6 RDC BANDWIDTH / TRACKING RATE SELECTION

Selection of the RDC bandwidth and tracking rate requires choosing the appropriate resistors and capacitors values (ie R_s, R_c, R_v, R_b, C_{b1} and C_{b2} shown in figure 5.18) with the aid of a *passive component selection* software provided by the RDC supplier.

The selected values of the passive components are:

$R_s = 30\ k\Omega$
$R_c = 30\ k\Omega$
$R_v = 60\ k\Omega$
$R_b = 460\ k\Omega$
$C_{bw} = 10000\ pF$
$C_{bw/10} = 1000\ pF$

The software was also used to compute the values of:

RDC bandwidth = 200 Hz
maximum tracking rate = 18 revs per sec.

5.6.3 CONNECTION OF RDC TO COMPUTER

In its 16 bit (resolution) mode the RDC provides 16 output data points for a computer 24-bit I/O interface card, as shown in figure 5.18, labelled pins 23 to 38 on the RDC chip. Two I/O bits of the 24-bit card where used as to trigger the output of the high and low bytes of the of the RDC (ie bits labelled EM and EL in figure 5.18).

In order not to sample the angular position (digital) while the converter is busy, the RDC chip is provided with pins CB and INB. Pin CB is usually OR-gated with a *dummy* bit supplied from the interface card, and the output of the gate is used to enable the INB pin as shown in figure 5.20. This ensures that the output data of the RDC is only valid after conversion.

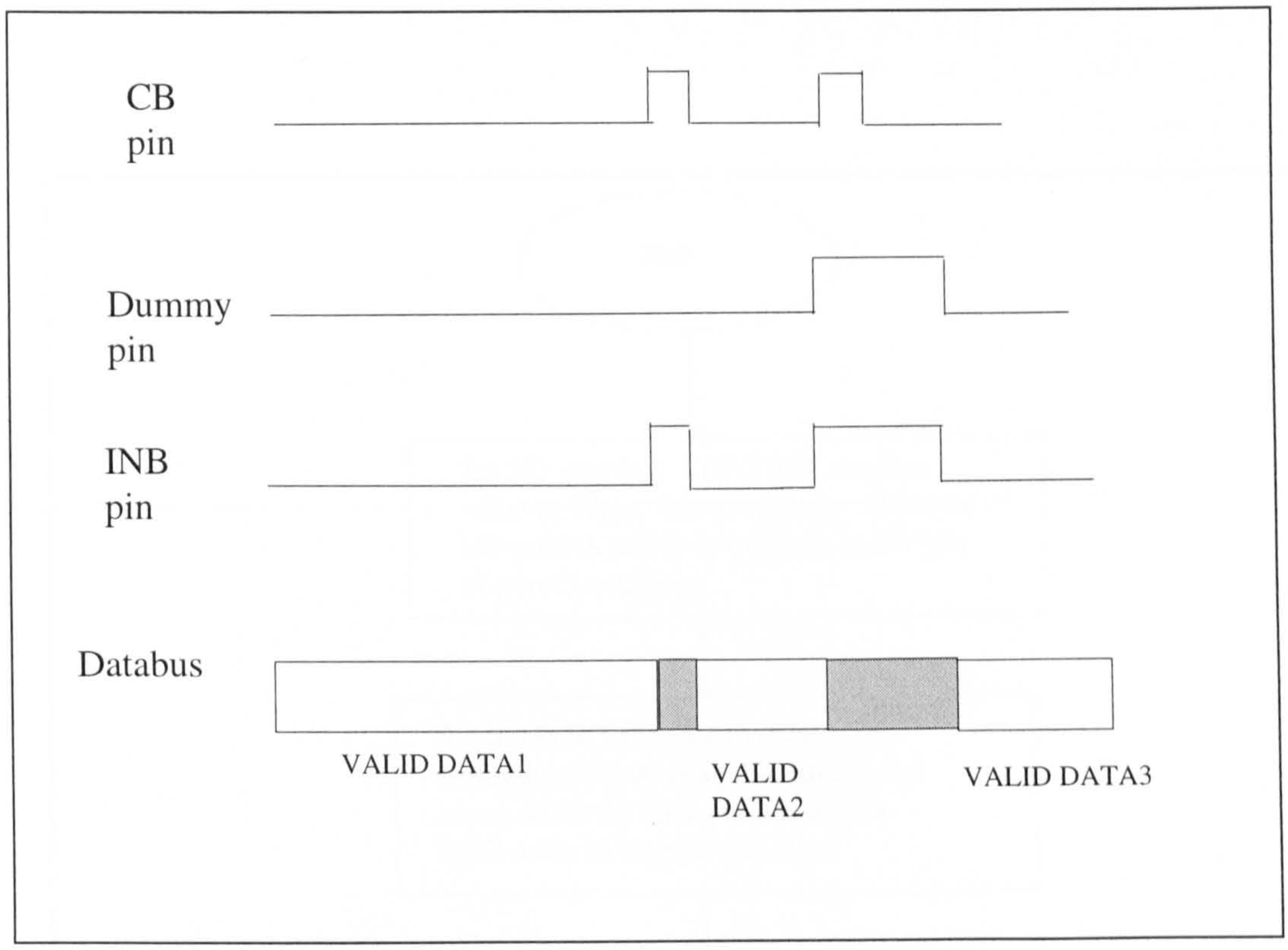


Figure 5.20: Timing diagram of RDC

PIN/STATUS	CB	DUMMY	OR-GATE OUTPUT (INB)
	0	0	0
	0	1	1
	1	0	1
	1	1	1

NB: PIN INB IS ACTIVE WHEN LOW

Table 5.17: Truth table of RDC

From the truth table shown in Table 5.17, when CB is LOW the INB pin is set LOW freezing the last data value in the data-bus read before the rising edge of CB, illustrated in the timing diagram shown in figure 5.24. The overall sequence of steps required for computer acquisition of angular position (digital word) is given in the flow chart depicted in figure 5.21

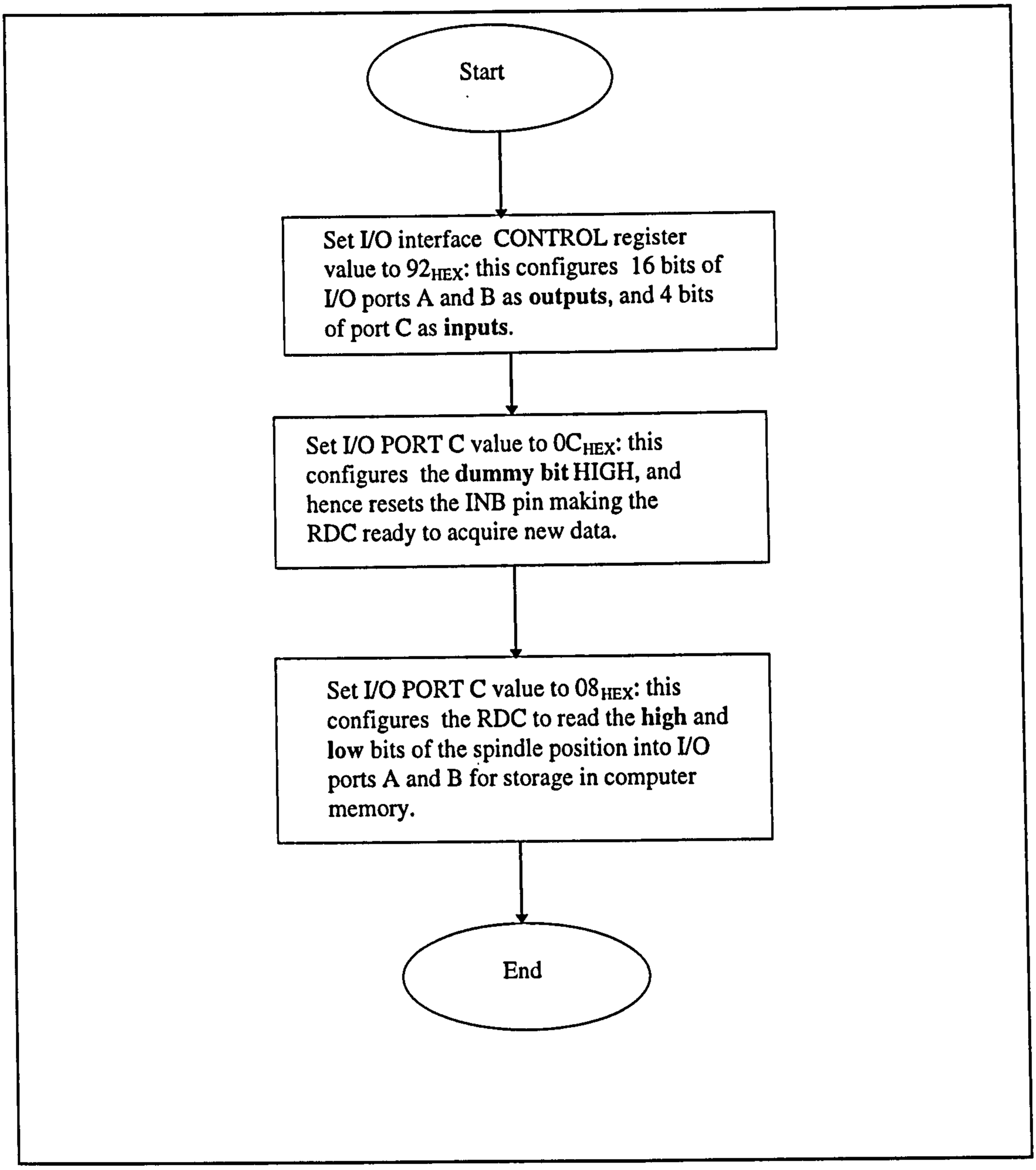


Figure 5.21: Flow chart showing the digital acquisition of spindle angular position

5.7 PC AND ITS INTERFACE CARD

A Viglen* 486DX personal computer was used for developing the spindle metrology software and for controlling the auto-adjustment mechanism. The computer

---

* Viglen Ltd, Viglen House, Alperton Lane, Middlesex, HA0 1DX

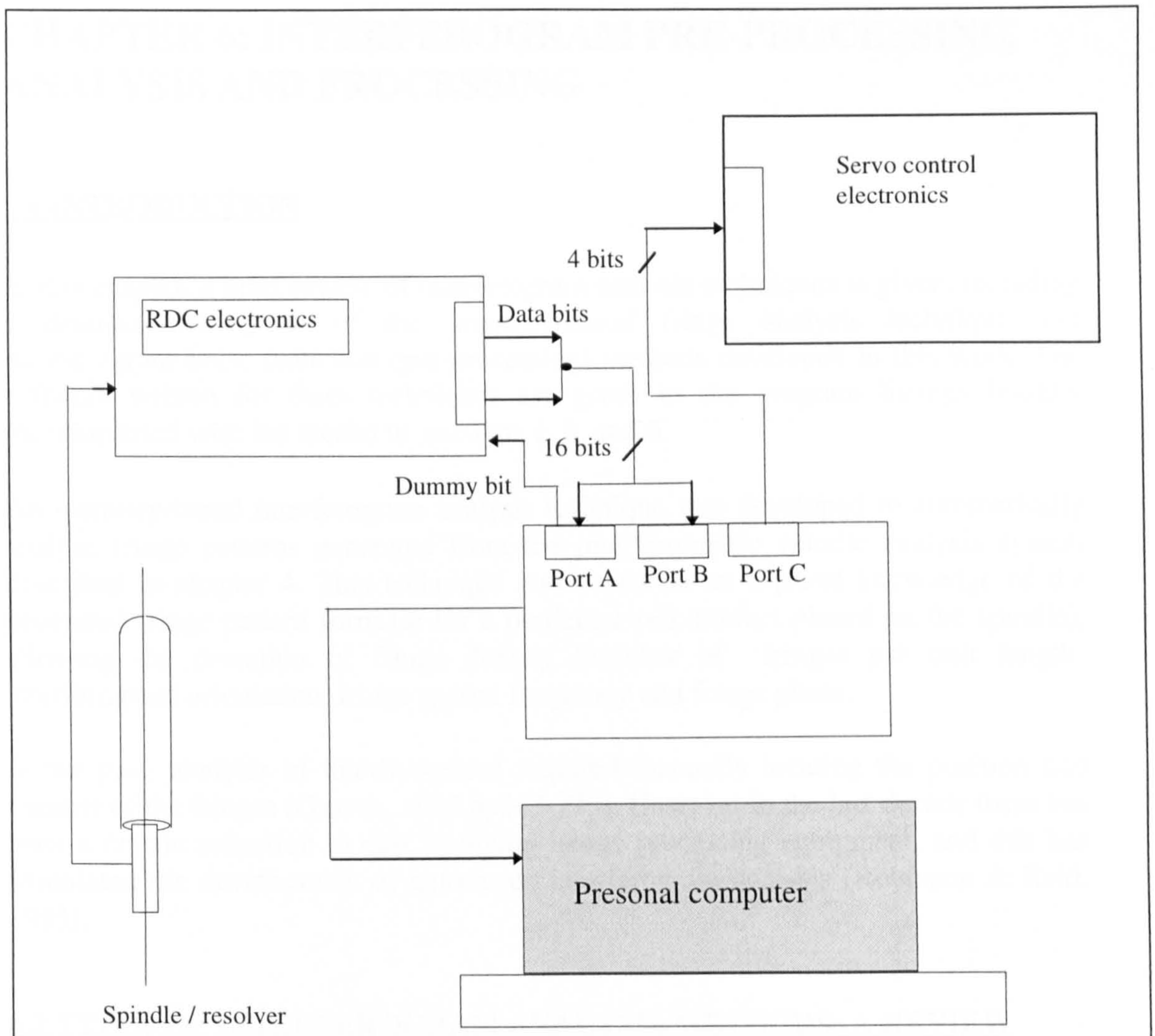


Figure 5.22: PC interface with servo and RDC electronics

incorporated an interface card (PC36 AT/LP) supplied by Amplicon** to provide the required computer - peripheral interface. This device has a capacity of 24 I/O lines and, depending on how it is programmed, a specified number of the lines are used as output and/or input lines. The board also provides up to five interrupt lines.

The 24 bit I/O card was used to interface the RDC, and interferometer servo electronics to a personal computer as shown schematically in figure 5.22.

** Amplicon Liveline Ltd, Centenary Industrial Estate, Holligdean Rd, Brighton, East Sussex, BN2 4AX

CHAPTER 6: INTERFEROGRAM PRE-PROCESSING, ANALYSIS AND PROCESSING

6.1 INTRODUCTION

In this chapter, a brief review of interferogram analysis techniques is given, including a detailed description of the *intensity-based* fringe analysis technique and interferogram noise reduction (pre-processing) methods developed in this work. The software written for thses techniques are given in the program listings booklet (accompanied with the thesis) in sections 1, 4 and 6.

An intensity-based interferogram analysis technique was developed to automatically analyse fringe patterns generated from the interferometric spindle analysis system described in chapter 4. This technique was based on an a priori knowledge of the generated fringe pattern form (ie for a particular *test* artefact placed on the spindle), allowing the detection of fringe density (number of fringes per unit length) interferogram orientation, fringe spatial frequency and fringe phase.

In the past, analysis of interferograms required manually locating the position and number of the fringes (Gasvik, 1995 pp247-273). However in the last decade there has been a drastic reduction in cost in digital image processing equipment, and this has stimulated the development of automated interferogram analysis (Robinson & Reid, 1993).

6.2 TYPES OF INTERFEROGRAM ANALYSIS METHODS: A REVIEW

There is a number of techniques for analysing fringe patterns (Reid, 1986) these can be categorised as:

- Intensity-based
- Temporal phase-measurement
- Spatial phase-measurement

Intensity-based interferogram analysis techniques involve the measurement of the fringe minimum (or maximum) points from intensity profile data. Prior to the measurement of fringe maxima (or minima), an interferogram is pre-processed; this involves the application of digital image processing to simplify and enhance the fringe intensity profile.

The equation which describes the intensity profile of a fringe pattern (ie derived in section 3.7.2) is:

$$I(X) = 2I_0(1 + \cos\Delta\phi) = a + b\cos(kX) \qquad (6.1)$$

where, I_0 : mean intensity
a, b : DC and AC intensity terms respectively

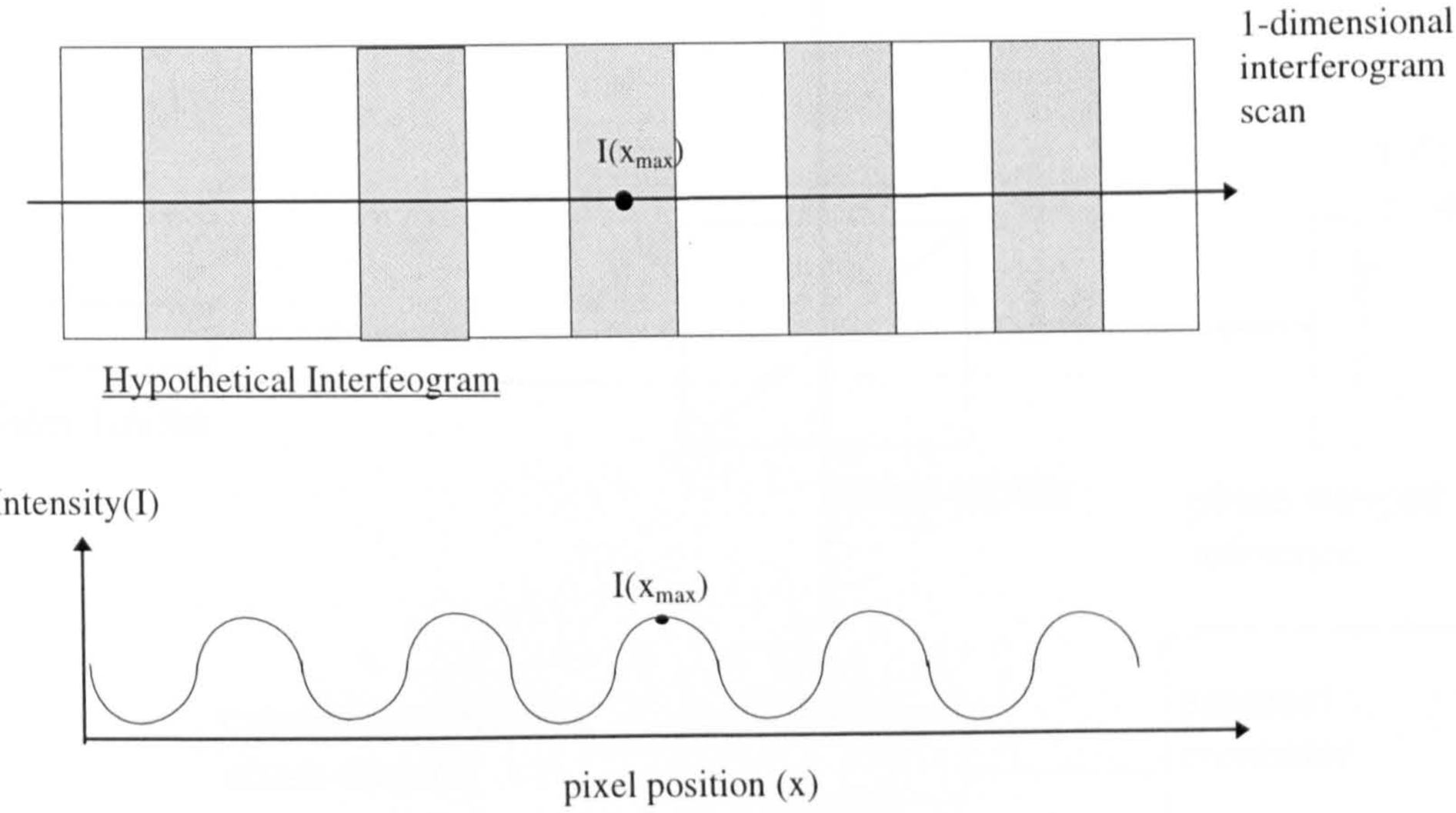


Figure 6.1: Fringe minima detection based on intensity measurements

$\Delta\phi$: phase change
k : wave number
X : spatial position along fringe pattern (fringe maximum is located at $X = X_{max}$)

The measurement of fringe-maxima position (X_{max}) involves the local detection of intensity maxima value $I(X_{max})$ given in equation 6.1 (as shown schematically in figure 6.1). In this work, interferogram maxima where detected to sub-pixel accuracy by fitting a polynomial of the form:

$$A_nX^n + A_{n-1}X^{n-1} + A_{n-3}X^{n-3} + A_{n-4}X^{n-4} + ... + A_0X^0 \qquad (6.2)$$

to fringe intensity data points, see section 6.5 for more details on this.

Yatagai (1993) described a typical hardware configuration for intensity based fringe analysis to consist of a high-resolution camera for interferogram acquisition, a frame-

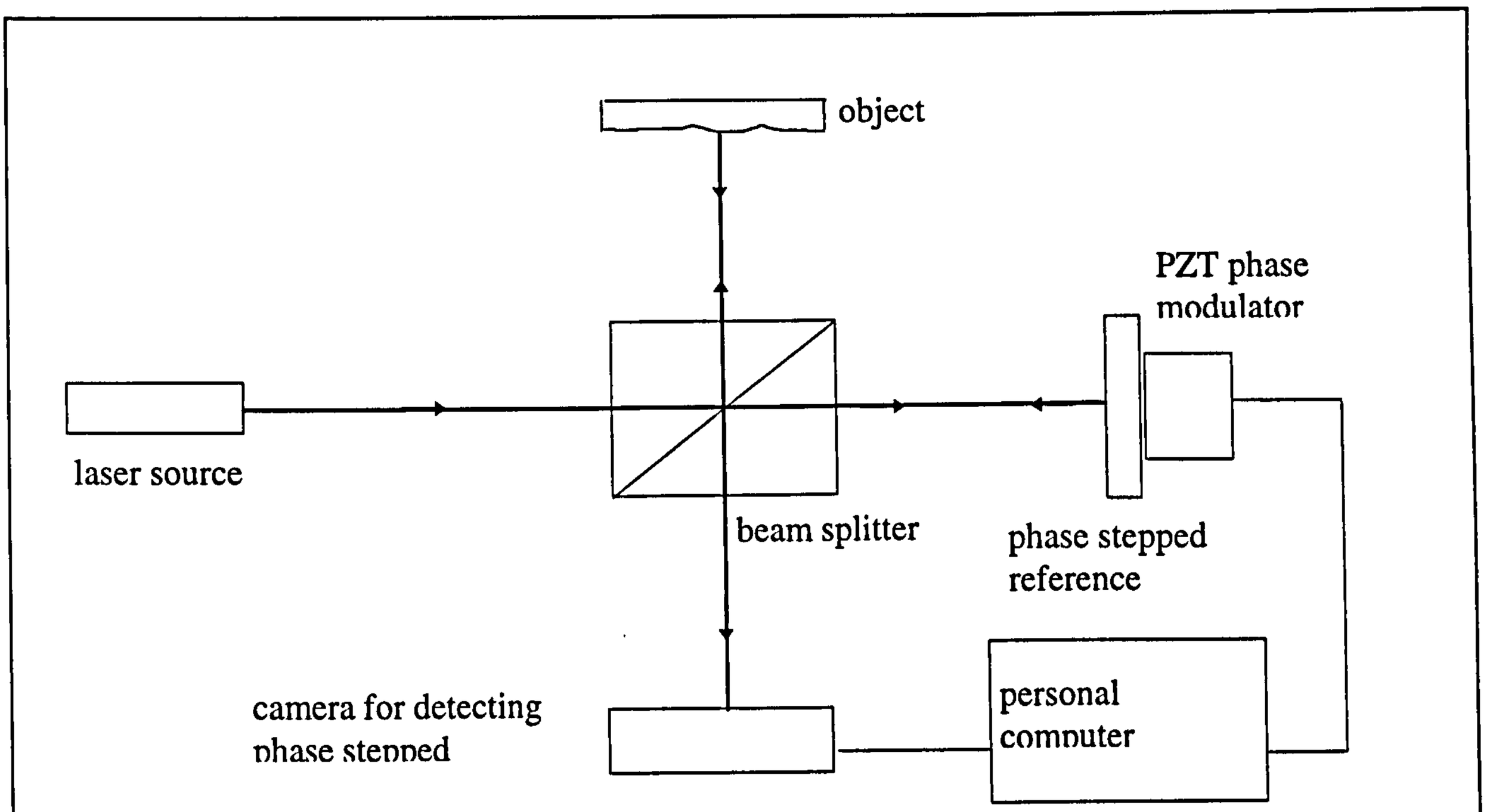


Figure 6.2: Interferometer set up for temporal phase measurement interferogram analysis after Creath (1993)

grabber to digitise the fringe pattern, a computer to process/store data, a monitor and a mouse/light-pen to manipulate pixels in the interferogram. Intensity-based analysis is a widely used technique for analysing interferograms due to its simplicity and low cost. However it is usually important to minimise the influence of noise in the interferogram (Gasvik, 1995 pp247-273) when using intensity-based methods.

Generally, temporal-phase interferogram analysis involves introducing a known phase-shift on the reference arm of an interferometer. There are two types temporal phase-measurement interferogram analysis methods. These are: *electronic heterodyne detection* and *digital* methods. A commercial example of the application of heterodyne detection (Wilson & Hawkes, 1989 pp366-367) is in the dual frequency Hewelett-Packard distance measurement interferometer (Slocum, 1992 pp190-191).

Digital temporal-phase methods usually require specified phase-shift (ie ramp function) or phase-step (ie step function) increments of the reference beam of an interferometer (Creath, 1993) as depicted in figure 6.2. As the reference beam is

changed by a known amount, interferogram frames are digitised and stored. A number of computer algorithms have been developed (Schwider et al 1983, Hariharan et al 1987, Surrel 1993 and Creath 1993) to calculate fringe phase (ϕ) and intensity (I) values of fringes from stored interferogram frames. For instance, the value of ϕ for a *three-frame* interferogram analysis (ie phase shifted in time at $\pi/4$, $\pi/2$ and $3\pi/4$ respectively) is:

$$\phi = \tan^{-1} \frac{I_{\pi/2} - I_{3\pi/4}}{I_{\pi/2} - I_{\pi/4}} \qquad (6.2)$$

where $I_{\pi/4}$, $I_{\pi/2}$ and $I_{3\pi/4}$ are interferogram frames after phase shifting the reference beam by $\pi/4$, $\pi/2$ and $3\pi/4$ respectively (Gasvik, 1995 pp247-273) .

In spatial phase interferogram analysis, fringe phase information is separated out in space (across the image plane) and is recorded in one frame-time (Kujawinska, 1993, Williams et al 1991). There two types of spatial phase-measurement techniques, these are: *phase-shifting (or phase-stepping)* and *spatial-carrier* methods.

In the phase-shifting method, the output of the interferometer is sub-divided into output *channels*. A typical application of phase-shifting interferogram analysis is carried out with a 3-channel interferometer (Gasvik, 1995 pp247-273) as shown in figure 6.3. In this interferometer two of the output channels are phase-shifted by a known amount with the aid of two quarter-wave plate polarisers. The output of the interferometer is detected in one frame time and the calculation of the 3-frame interferogram phase is:

$$\phi = \tan^{-1} \frac{2I_{\pi/2} - I_{\pi} - I_{1}}{I_{\pi} - I_{1}} \qquad (6.3)$$

In the spatial-carrier method, a carrier fringe-pattern is superimposed onto the analysed interferogram, usually by tilting the reference beam of the interferometer (Reid, 1986). This gives a resultant fringe pattern of intensity profile:

$$g(x,y) = a(x,y) + b\cos\{\phi(x,y) + 2\pi f_0 x\} \qquad (6.4)$$

where, a: mean intensity of partial interference signals
b: interference term
ϕ: required phase measurement
f_0: spatial frequency of fringe pattern
x,y: spatial co-ordinates

This signal may be processed in the frequency domain using the fourier transform method developed by Takeda et al (1982), giving the interferogram phase of:

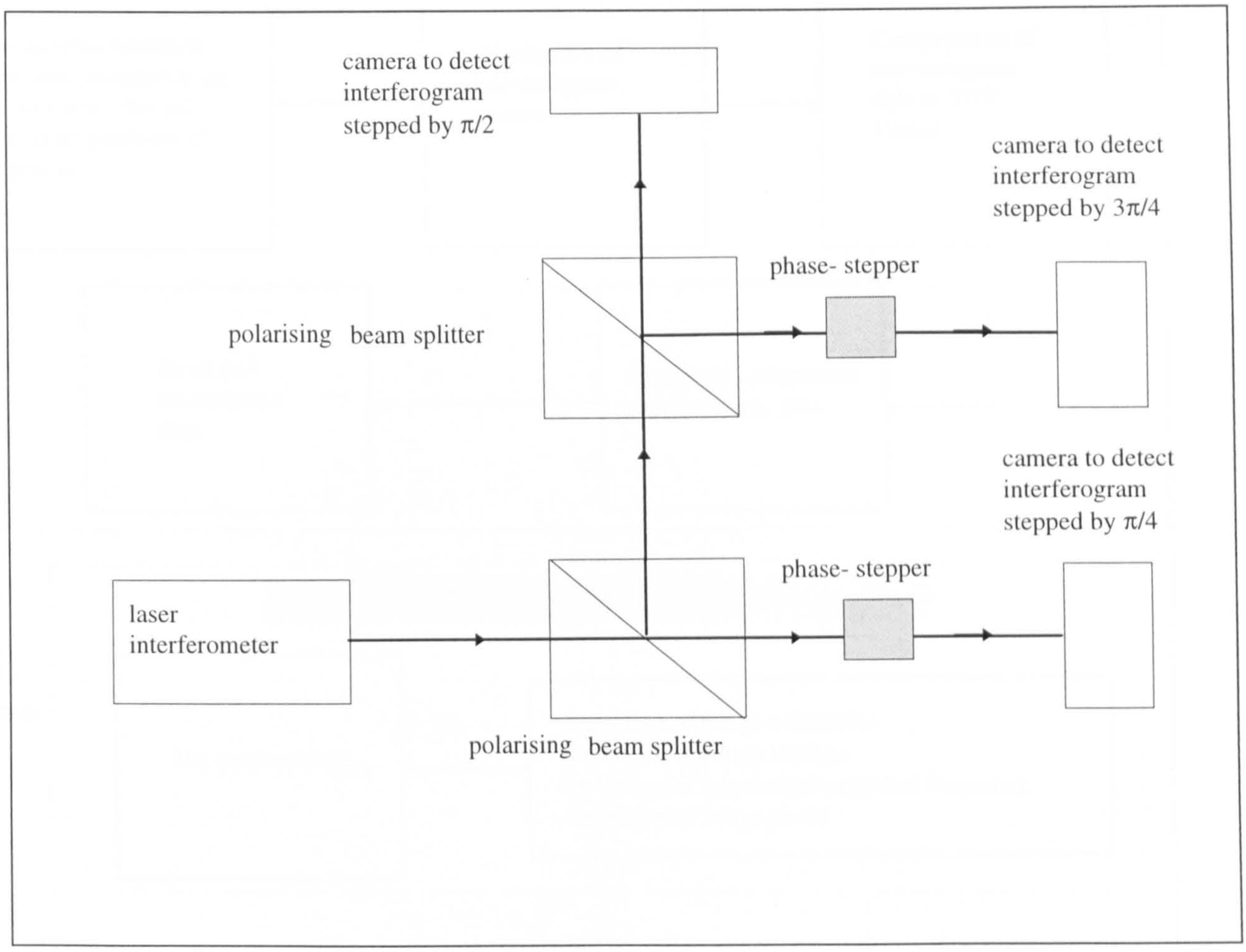


Figure 6.3: Interferometer set up for spatial phase measurement interferogram analysis after Kujawinska (1993)

$$\phi(x,y) = \tan^{-1} \frac{\mathrm{Im}\, c(x, y)}{\mathrm{Re}\, c(x, y)} \qquad (6.5)$$

$$\text{where } c = 0.5b(x,y)e^{i\phi(x,y)} \qquad (6.6)$$

According to Creath (1993), advantages of (temporal/spatial) phase-measurement methods over intensity-based techniques are:

- Precision of phase-measurement techniques is a factor of 10 to a 100 greater than intensity-based techniques

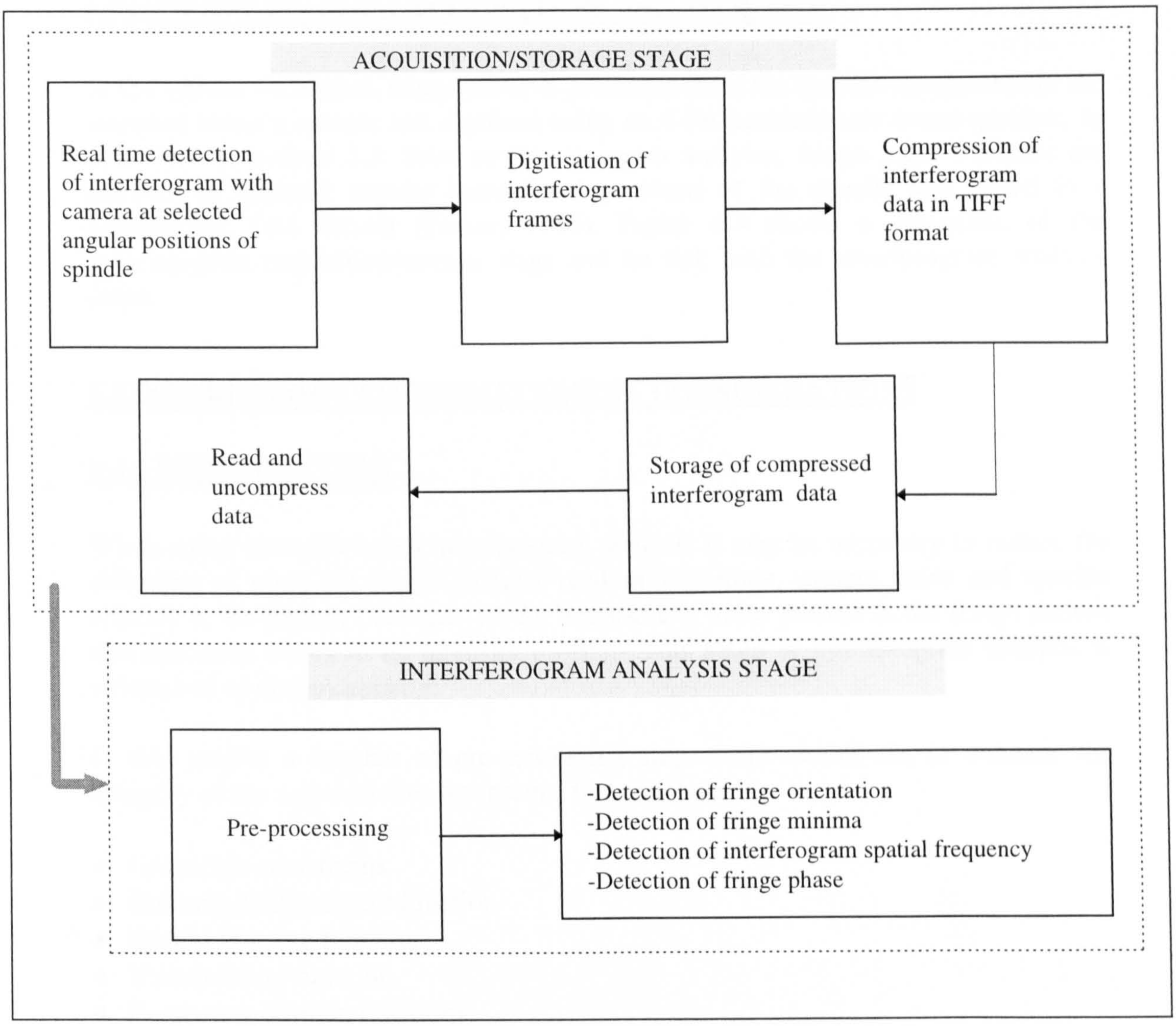


Figure 6.4: Interferogram acquisition/storage and analysis

- Phase-measurement techniques provide very high resolution and repeatable data to a thousandth of a wavelength

Phase-measurement techniques are generally suited for interferometric measurement of *shape* and *surface form* where very high resolution *phase-maps* are generated. Metrological data is extracted from the phase-maps using phase unwrapping algorithms (Huntley 1989) which usually require very high computational power. On the other hand, the intensity-based interferogram algorithm developed in this project relies on the measurement of fringe-maximum spatial co-ordinates and intensity of a specified area of the sampled interferogram. This makes the algorithm simple to use, and computationally efficient.

6.3 INTERFEROGRAM ACQUISITION AND STORAGE

In the system developed, fringe patterns generated from the spindle interferometer are acquired using a camera and digitised using an 8-bit monochrome frame-grabber, as described in section 5.3. Prior to interferogram analysis, fringe pattern frames are sampled at selected angular incremental positions of the spindle and stored in a compressed *TGA* format (Parker, 1995). Figure 6.4 shows a schematic of the interferogram acquisition/storage stage and its link with the interferogram analysis stage.

6.4 INTERFEROGRAM ENHANCEMENT CONSIDERATIONS

6.4.1 INTRODUCTION

When using intensity-based interferogram analysis it may be necessary to reduce the influence of noise (ie fringe contrast random variations, camera noise and speckle effects) in the pattern (Yatagai, 1993), because any noise present in the fringe pattern can introduce errors in its intensity profile. This stage of interferogram analysis is referred to as *pre-processing*.

In this project a number of pre-processing steps were carried-out to enhance the integrity of the acquired interferograms, these include:

- Geometric transforms
- Contrast enhancement function
- Spatial filtering function
- Thresholding function
- Frequency domain functions
- Morphological functions

6.4.2 GEOMETRIC TRANSFORMS

Geometric transforms involve the modification of interferogram pixel position, in order to:

- improve the detection of details in the fringe pattern
- to orientate the fringe pattern in the vertical direction
- correct interferogram distortions due to camera lens

According to Sonka et al (1993) a geometric transform is a vector $\overline{T}$ that maps an image pixel (x, y) to a new position (x', y'), the transformed co-ordinates are defined as:

$$x' = \overline{T}_x(x,y) \quad \text{and} \quad \overline{T}_y(x,y) \qquad (6.7)$$

respectively. A geometrically transformed interferogram O(x',y') is usually an approximation of the original I(x,y) where every pixel in O(x',y') is an estimate based on I(x,y).

Geometric transforms used in this project were:

Windowing
Rotation
Masking

The following sub-sections describe the application of these transforms in this work.

6.4.2.1 WINDOWING

Windowing was used to extract a rectangular portion referred to as an *area of interest* (AOI) from interferogram I(x,y) as shown schematically in figure 6.5. The AOI is specified by its upper-left (x_L, y_L) and lower-right (x_R, y_R) corners. Figures 6.6 and 6.7 show an interferogram before and after *windowing* respectively.

6.4.2.2 ROTATION

A rotation transform is used to orientate an acquired interferogram in the vertical direction, in order to simplify the process of fringe maxima detection. This function is specified by interferogram *orientation* (θ) (ie derived from the Fourier transform technique described in section 6.4.6.6) and *centriod* (x_0,y_0). Figure 6.8 shows a schematic of the rotation of interferogram I(x,y) based the Fourier transform technique.

The rotation of I(x,y) is expressed mathematically (Visilog, 1993) as:

$$\mathbf{ROT}\{I(x,y)\} = O(x',y') \qquad (6.8)$$

$$x' = (x-x_0)\cos\theta - (y-y_0)\sin\theta + x_0 \qquad (6.9)$$

$$y' = (x-x_0)\sin\theta + (y-y_0)\cos\theta + y_0 \qquad (6.10)$$

Where, x, y : are the co-ordinates of the interferogram before rotation
x', y' : are the co-ordinates of the interferogram after rotation
O(x',y'): is thetransformed interferogram
ROT : is the rotation transform sign

Figures 6.9 and 6.10 show an acquired interferogram before and after rotation respectively.

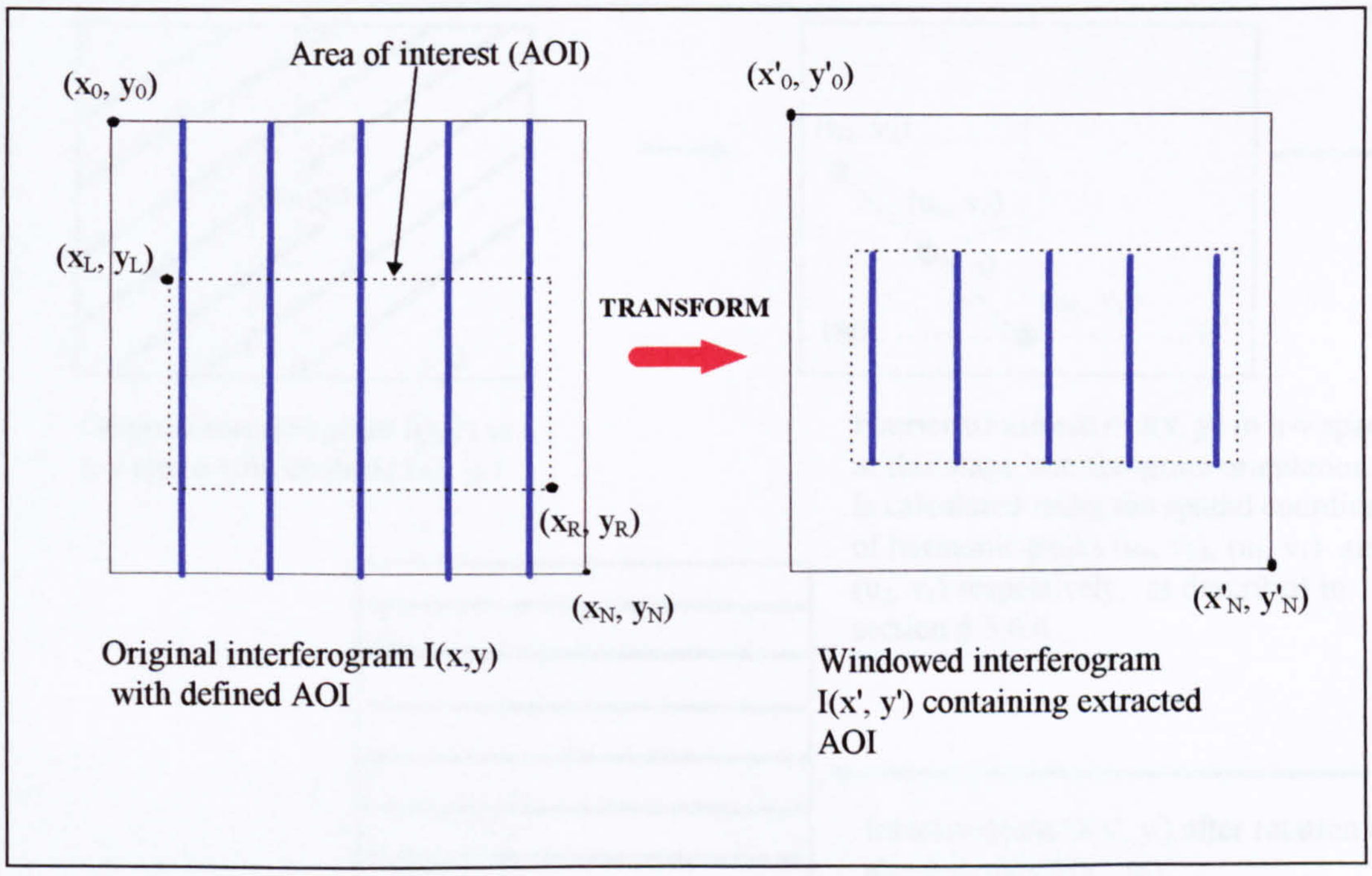
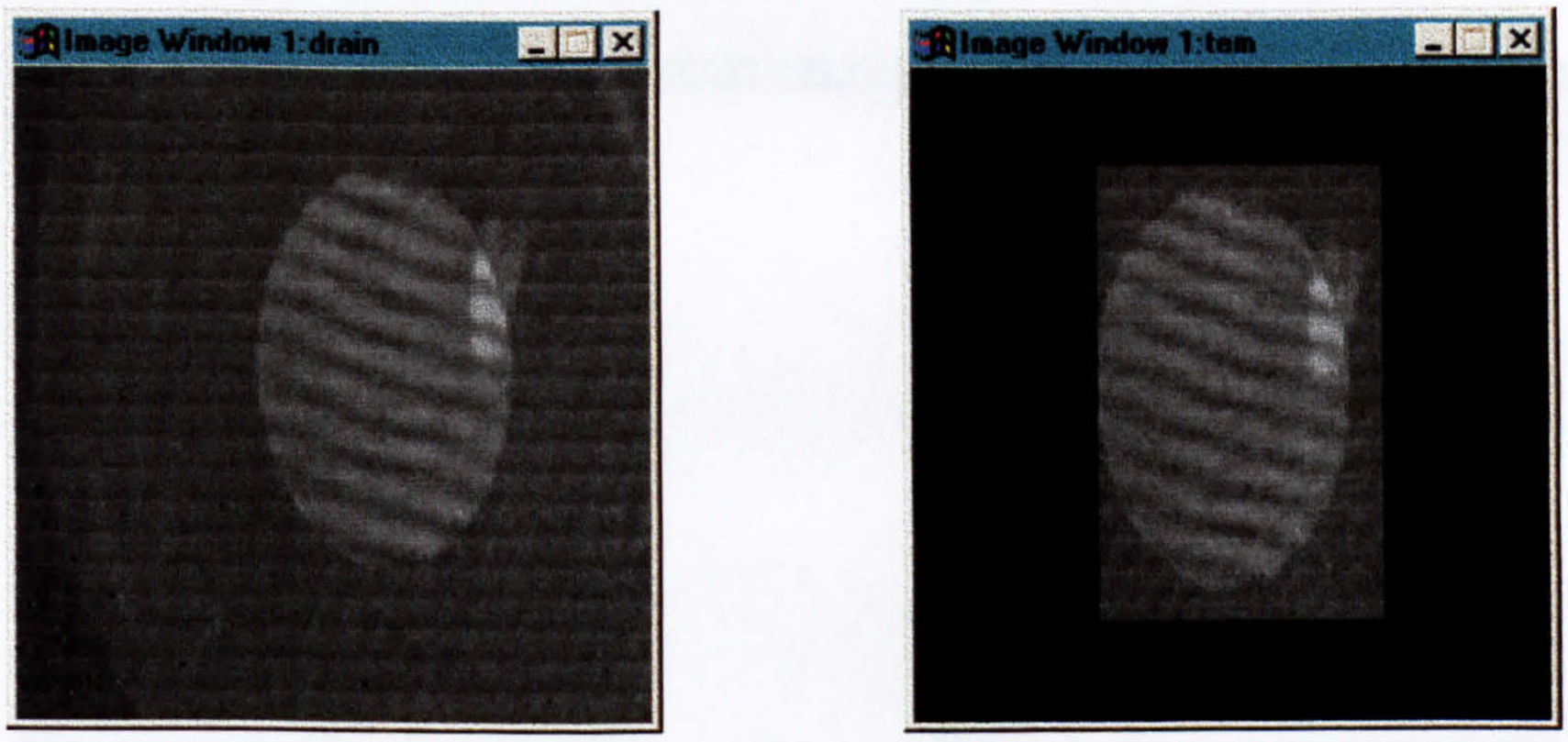


Figure 6.5: Schematic description of windowing



Figures 6.6 and 6.7: Full interferogram frame (256 × 256) before windowing and Interferogram after windowing (AOI is 58 × 90)

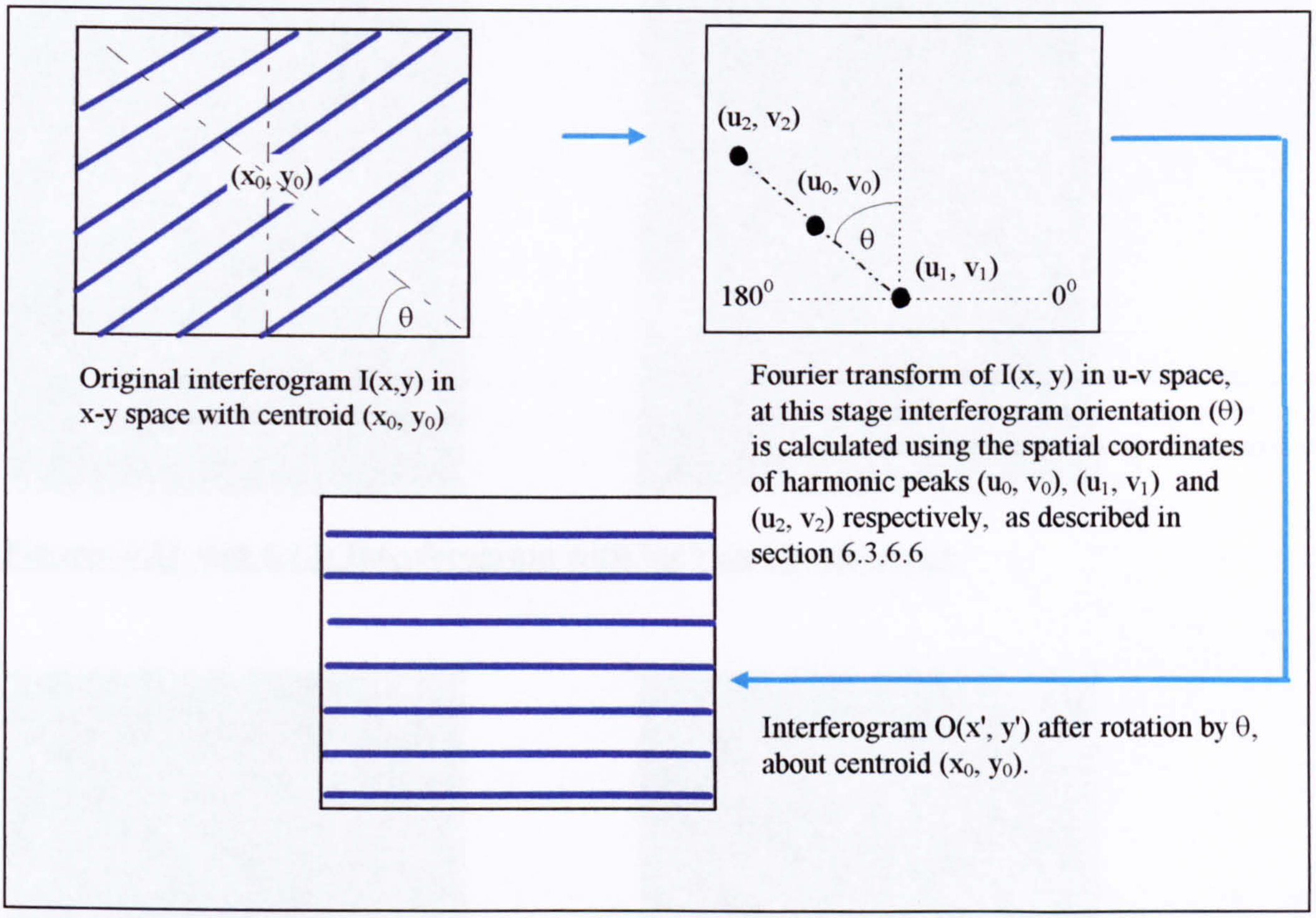
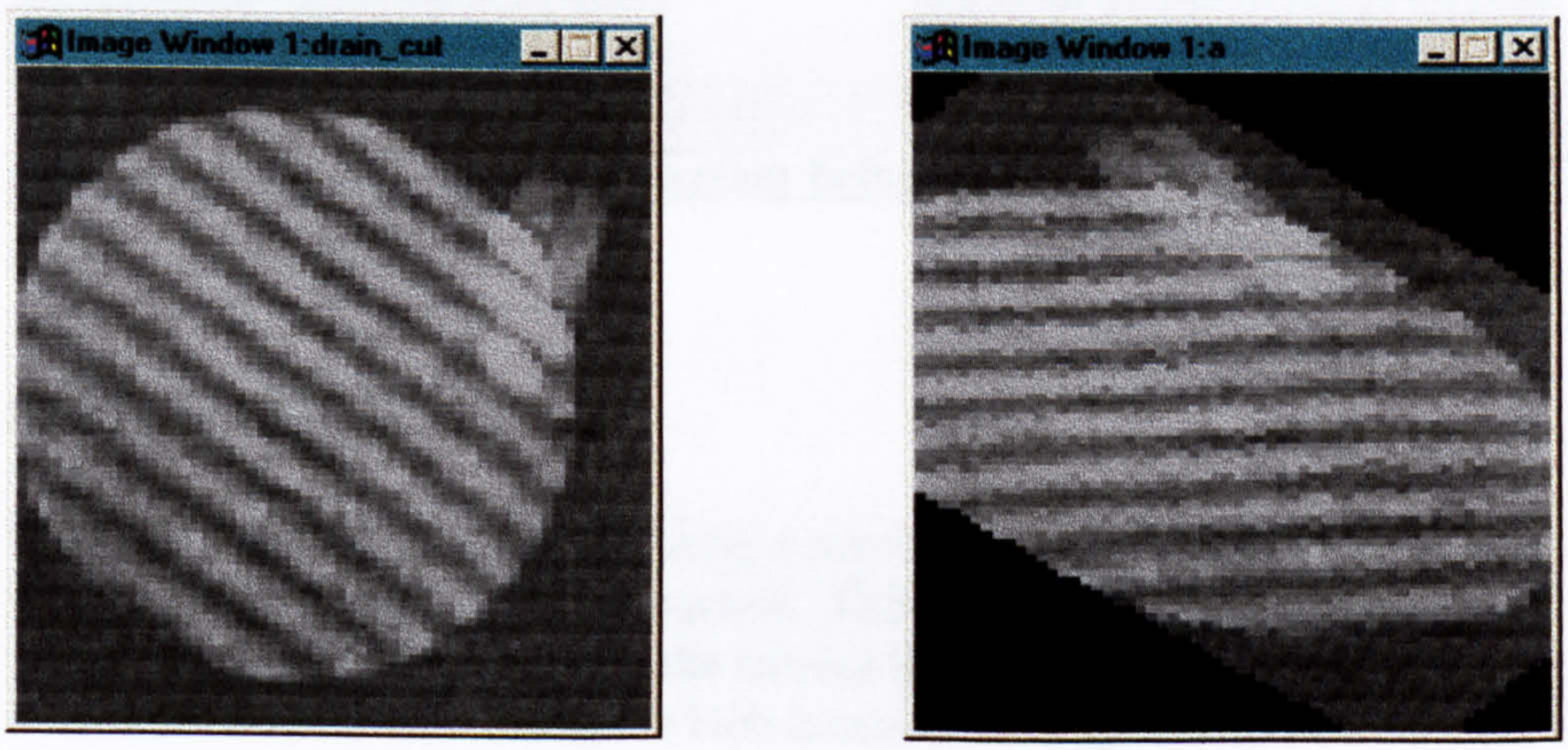
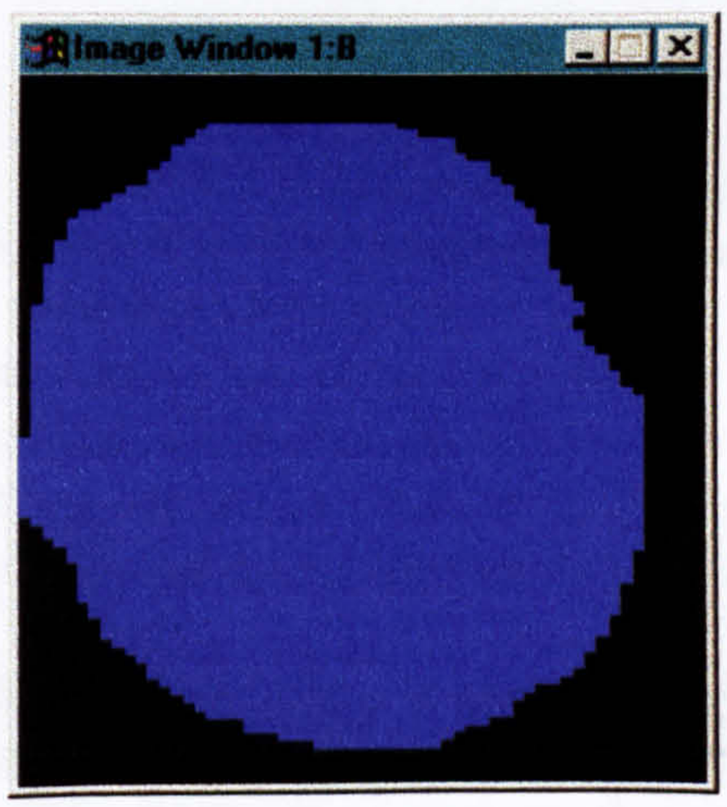


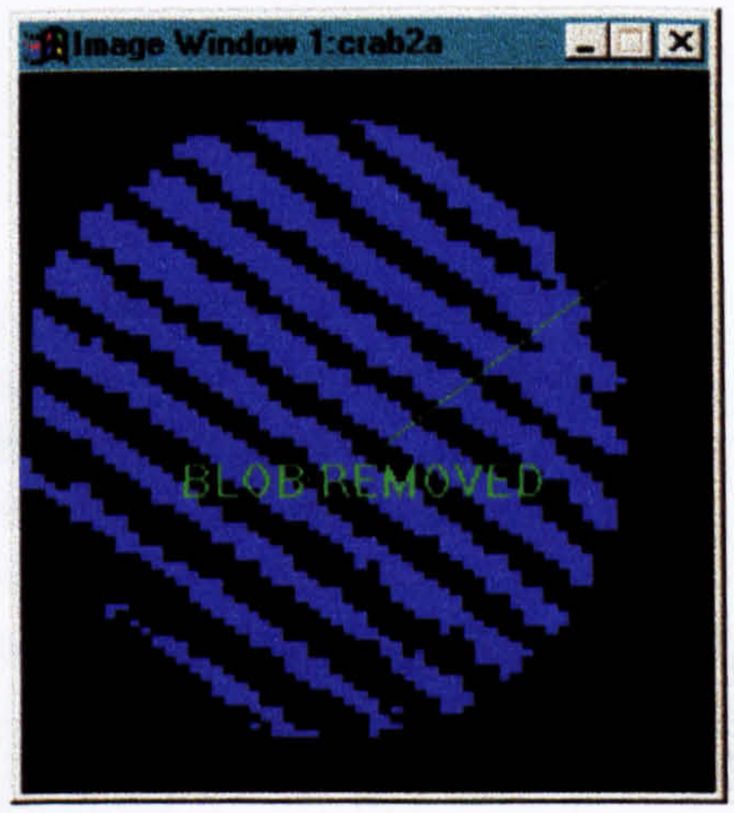
Figure 6.8: Schematic of interferogram rotation based on Fourier transform technique



Figures 6.9 and 6.10: Interferogram before after rotation by $\theta = 51.22^0$ at $(x_0, y_0) = (28, 45)$



Figures 6.11 and 6.12: Interferogram with blob and mask image



Figures 6.13 and 6.14: Interferogram before and after masking

6.4.2.3 MASKING

Masking was used to remove (mask) a particular noisy feature (ie resembling a *blob*), present in the acquired interferograms. The blob-like feature was introduced to the interferograms due to a mark on the camera lens. Figure 6.11 shows an acquired grey-level interferogram containing the blob located at the upper-right corner of the frame.

The *masking* function is essentially a logical AND operation of interferogram I(x,y) with a defined mask image M(x,y). The mask image relates to the preferred geometry of the interferogram (ie without the blob as depicted in figure 6.12), and is stored as a file on the hard-disk. Figures 6.13 and 6.14 show a binarised interferogram before and after masking respectively.

6.4.3 CONTRAST ENHANCEMENT

The contrast of the acquired interferogram frames was relatively poor. Figures 6.15 and 6.16 show a typical interferogram acquired during the rotation of the spindle, and its grey-level histogram respectively. The grey-level histogram is a chart used for quantifying the number of pixels per grey-level in an image, a statistic used for validating the quality of an image. As a result of camera non-linearity and variations in ambient lighting, the complete range of grey-levels (ie 0 to 256) was not used during interferogram acquisition (see figures 6.15 and 6.16), causing its poor contrast.

In order to enhance the contrast of the interferogram, the complete range of grey-values must be utilised, using a technique known as *contrast stretching* or sometimes referred to as *normalising* (Gasvik, 1995 pp236 - 237). The principle of normalising is to stretch the range of grey-levels of the interferogram, in order to get its contrast closer to its optimal value. The equation used for the contrast stretching routine (Visilog, 1993) is:

$$O(x,y) = \left(\frac{d-c}{b-a}\right)\left(I(x,y)-a\right) \qquad (6.11)$$

where, I(x,y), O(x,y): input and output interferogram data
d : maximum grey-scale value of interferogram after normalisation
c : minimum grey-level value of interferogram after normalisation
b : maximum grey-level value of interferogram before normalisation
a : minimum grey-level value of interferogram before normalisation
x, y : spatial co-ordinates of interferogram frame

Applying this routine to the interferogram shown in figure 6.15 improves its contrast (see figure 6.17), figure 6.18 illustrates this graphically. A flow chart of the developed contrast stretching routine is shown in figure 6.19.

6.4.4 SPATIAL FILTERING

A spatial filter routine was used to suppress the *grainy* nature (high spatial frequencies) of the acquired interferograms. High spatial frequency noise in interferograms which manifests itself as large variations in grey-levels from pixel to pixel (Gasvik, pp238 - 239), usually make the detection of fringe minimum (or maximum) points very difficult. A number of workers involved in the intensity-based interferogram analysis have developed spatial filtering algorithms to suit interferograms of different fringe spacing (Robinson & Williams 1986, Yatagai et al 1983).

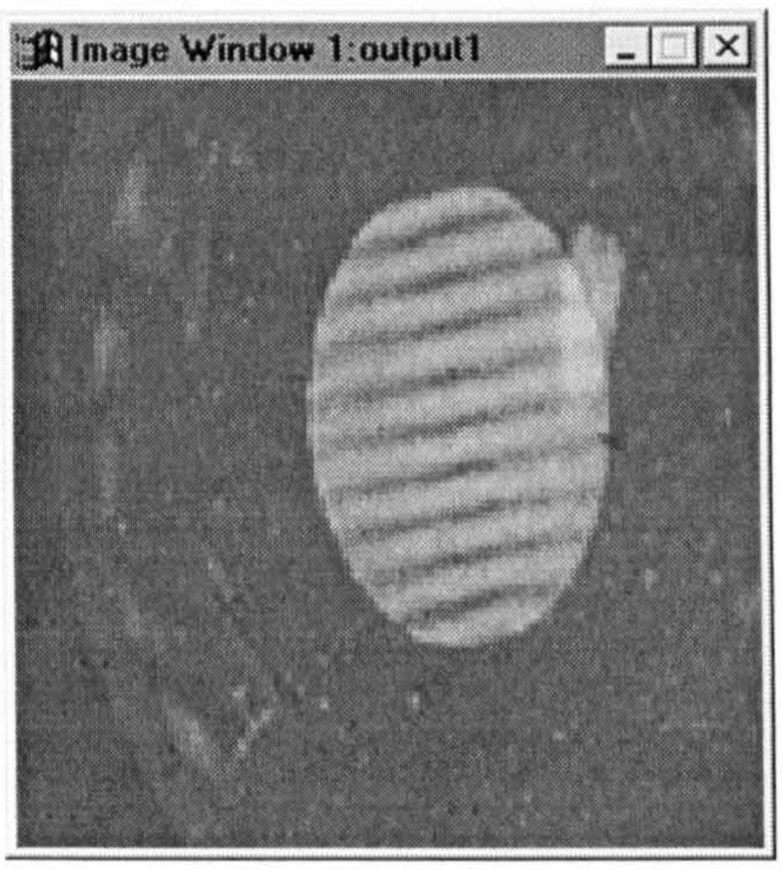


Figure 6.15: Interferogram before contrast stretching

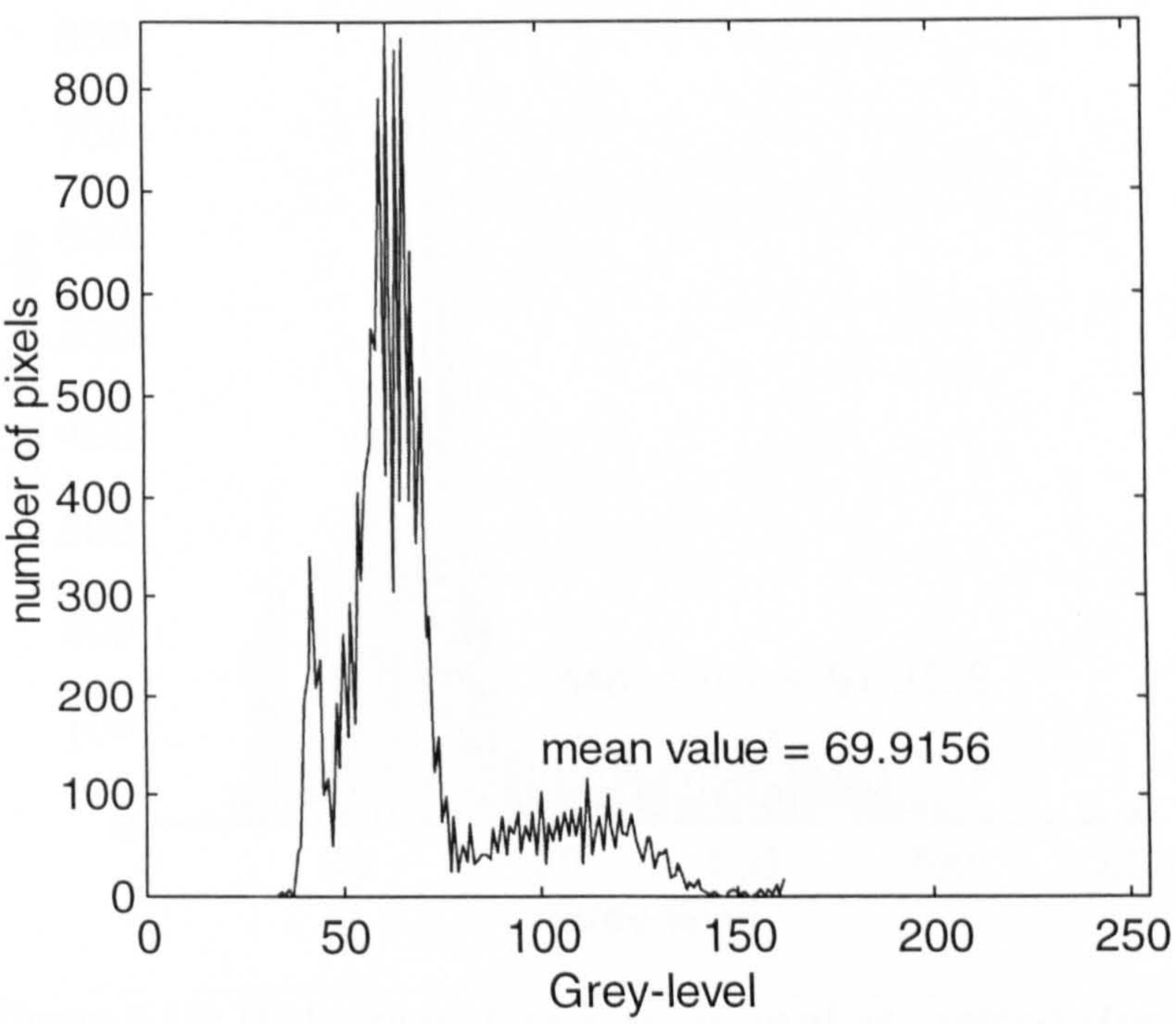


Figure 6.16: Histogram of the interferogram before contrast stretching

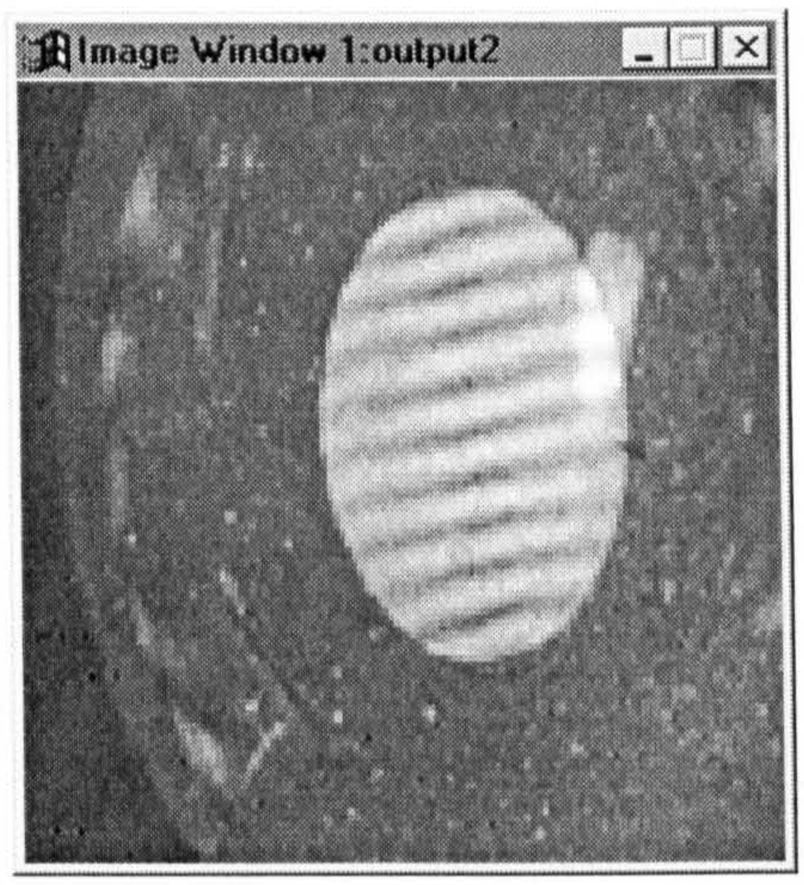


Figure 6.17: Interferogram after contrast stretching

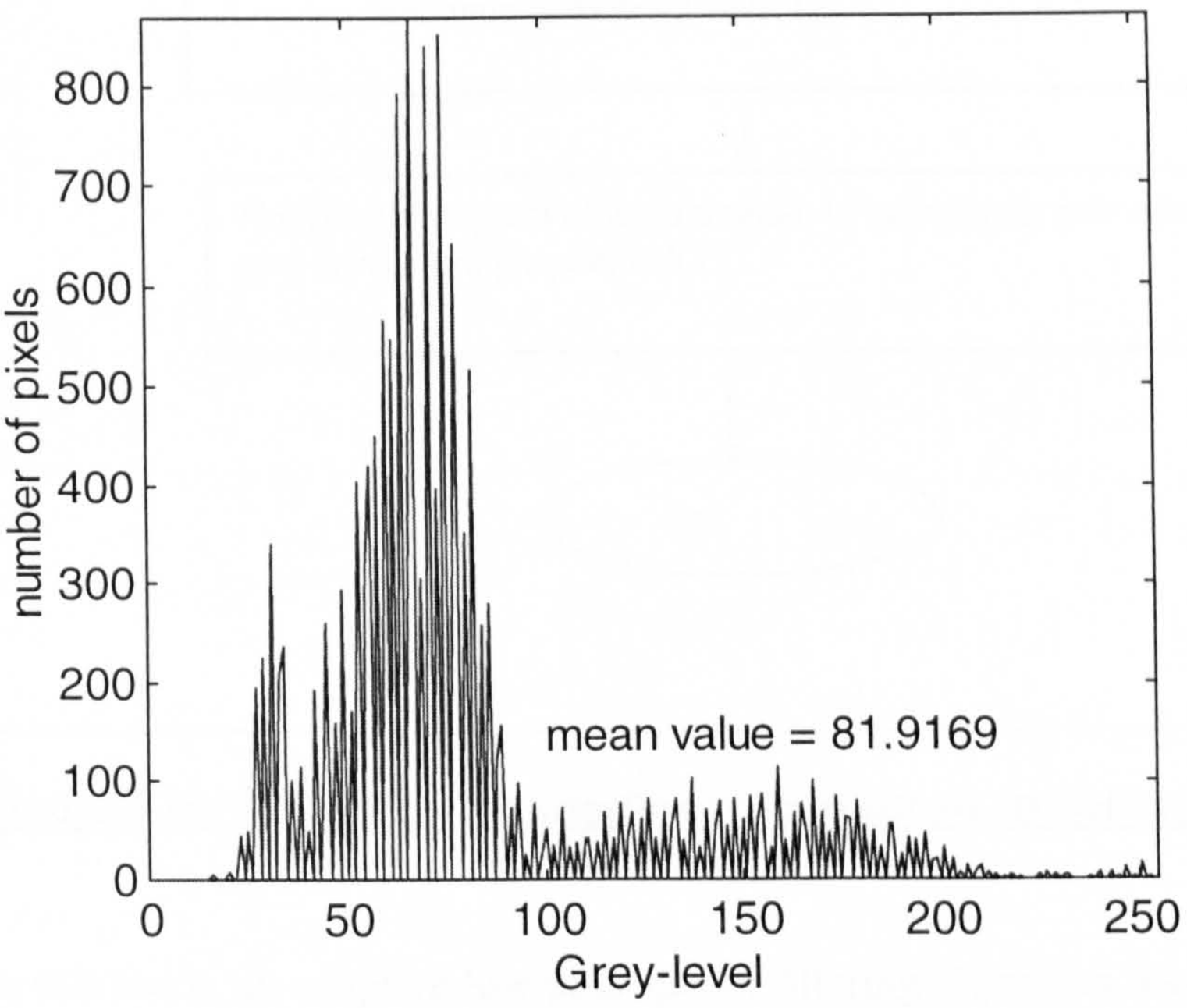


Figure 6.18: Histogram of interferogram after contrast stretching

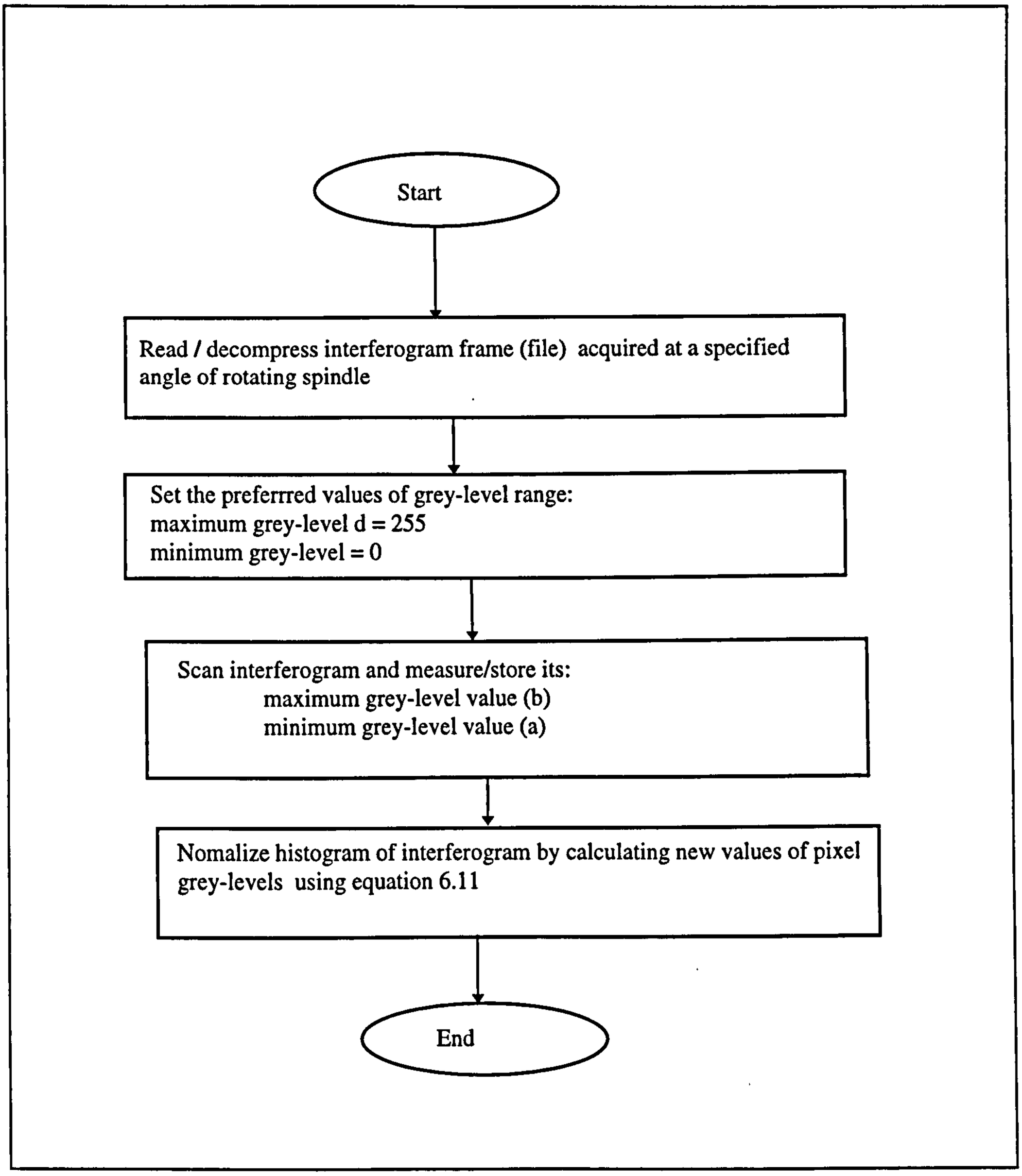


Figure 6.19: Algorithm for interferogram contrast stretching

In this work, an adaptive low pass spatial filtering algorithm was developed to *smooth* sampled interferograms, using a filter window whose size (ie matrix) is matched to the fringe density (ie which ranges from a 3×3 matrix for high fringe density, to a 5×5 size for low fringe density). The selected filter window is superimposed upon the analysed 58 by 90 fringe pattern area of interest and shifted pixel by pixel along it.

This filtering function is essentially a convolution of the input interferogram and the filter window, where each pixel of the input data is multiplied by the window values. The convolution spatial filtering function is expressed mathematically in its discrete form (Awcock & Thomas, 1995) as:

$$O(x,y) = I \otimes H = \sum_{m} \sum_{n} I(x,y) H(x-m, y-n) \qquad (6.12)$$

Where I : input interferogram
O : output interferogram after spatial filtering
H: filter window
x,y: number of rows and columns of interferogram frame
n.m: number of rows and columns of filter window
$\otimes$: convolution sign

The 3 by 3 and 5 by 5 filter window values used are:

$$\begin{bmatrix} 0.25 & 0.38 & 0.25 \\ 0.38 & 1 & 0.38 \\ 0.25 & 0.38 & 0.25 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 0.25 & 0.33 & 0.38 & 0.33 & 0.25 \\ 0.33 & 0.50 & 0.61 & 0.50 & 0.33 \\ 0.38 & 0.61 & 1 & 0.61 & 0.38 \\ 0.33 & 0.50 & 0.61 & 0.50 & 0.33 \\ 0.25 & 0.33 & 0.38 & 0.33 & 0.25 \end{bmatrix} \quad \text{receptively.}$$

These values are known as *weights* and they are used to retain required spatial frequencies of the input interferogram in a proportion related to the values of the respective windows. The weights given above relate to an exponential filter function (Visilog, 1993), and each weight in the respective window will modify the intensity of the interferogram as the window is moved across it, from one pixel to another.

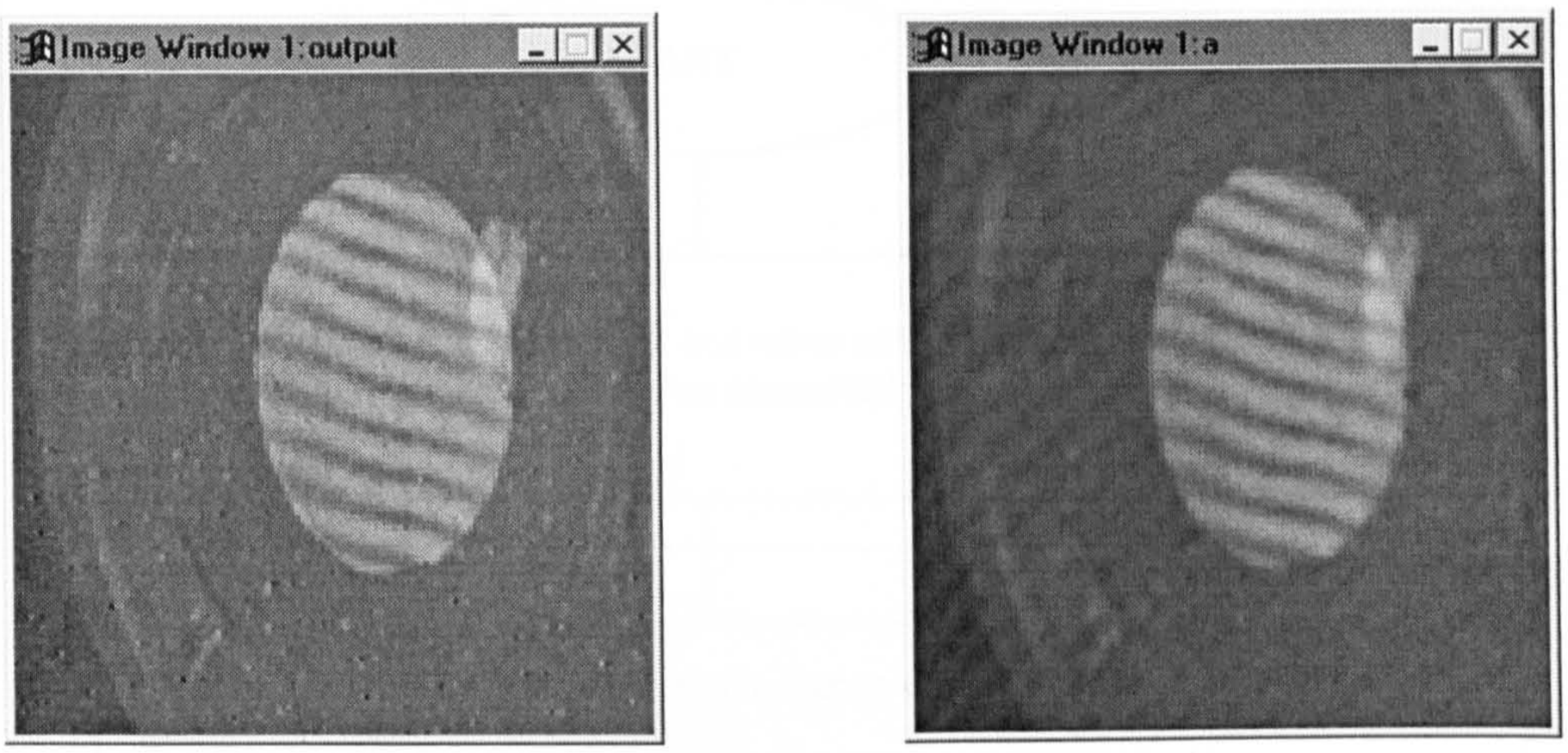
Figures 6.20a and 6.20b show an interferogram frame before and after the application of a 3 by 3 spatial filter respectively. The interferogram is averaged by the application of the spatial filter, this is further confirmed by its grey-level histogram before and after applying the filter shown in figures 6.21a and 6.21b respectively. The algorithm developed for the adaptive spatial filter is described with the aid of a flow diagram (shown in figure 6.22).

6.4.5 INTERFEROGRAM THRESHOLDING

A pre-processing technique was developed to transform the fringe pattern from grey-level to bi-level. This technique is known as *thresholding* or sometimes referred to *binarisation* (Visilog, 1993). Thresholding involves setting the grey-level values of an interferogram below a certain threshold (α) to 0 (ie black), and the remaining values α to 1 (ie white), as illustrated graphically in figure 6.23. Thresholding was used in this work to simplify fringe pattern intensity profile, so that the fringe peaks are easily detectable.

Figures 6.24a and 6.24b are the intensity profiles of the interferograms shown in figures 6.25a and 2.25b, before and after thresholding respectively. The algorithm used for interferogram thresholding is :

$$\underline{\text{IF}}\ I(x,y) \leq \alpha\ \underline{\text{THEN}}\ O(x,y) = 0$$
$$\underline{\text{ELSE}}\ O(x,y) = 1 \qquad (6.13)$$



Figures 6.20(a, b) : Interferogram (a) before and after (b) spatial filtering

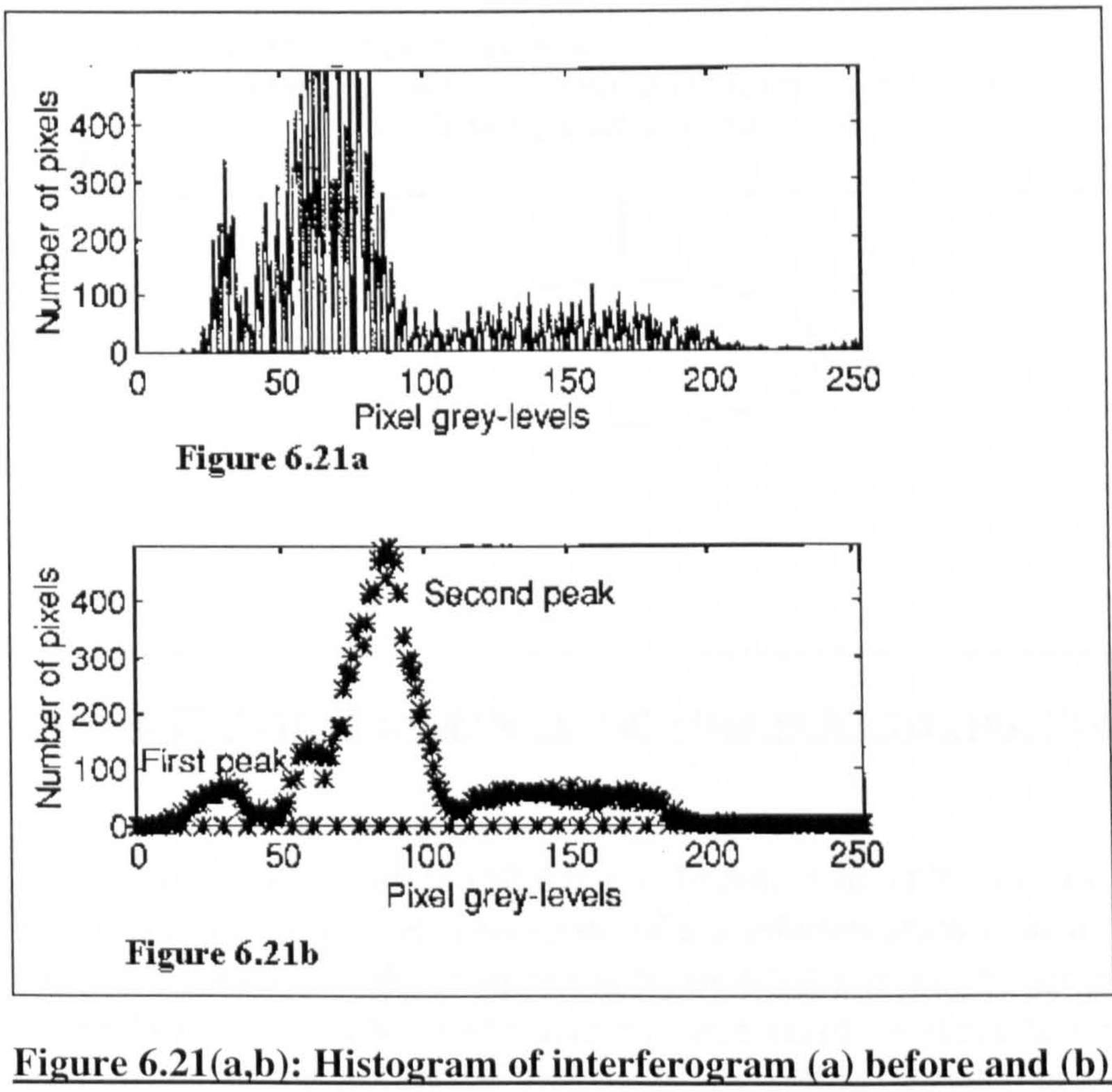


Figure 6.21a

Figure 6.21b

Figure 6.21(a,b): Histogram of interferogram (a) before and (b) after spatial filtering

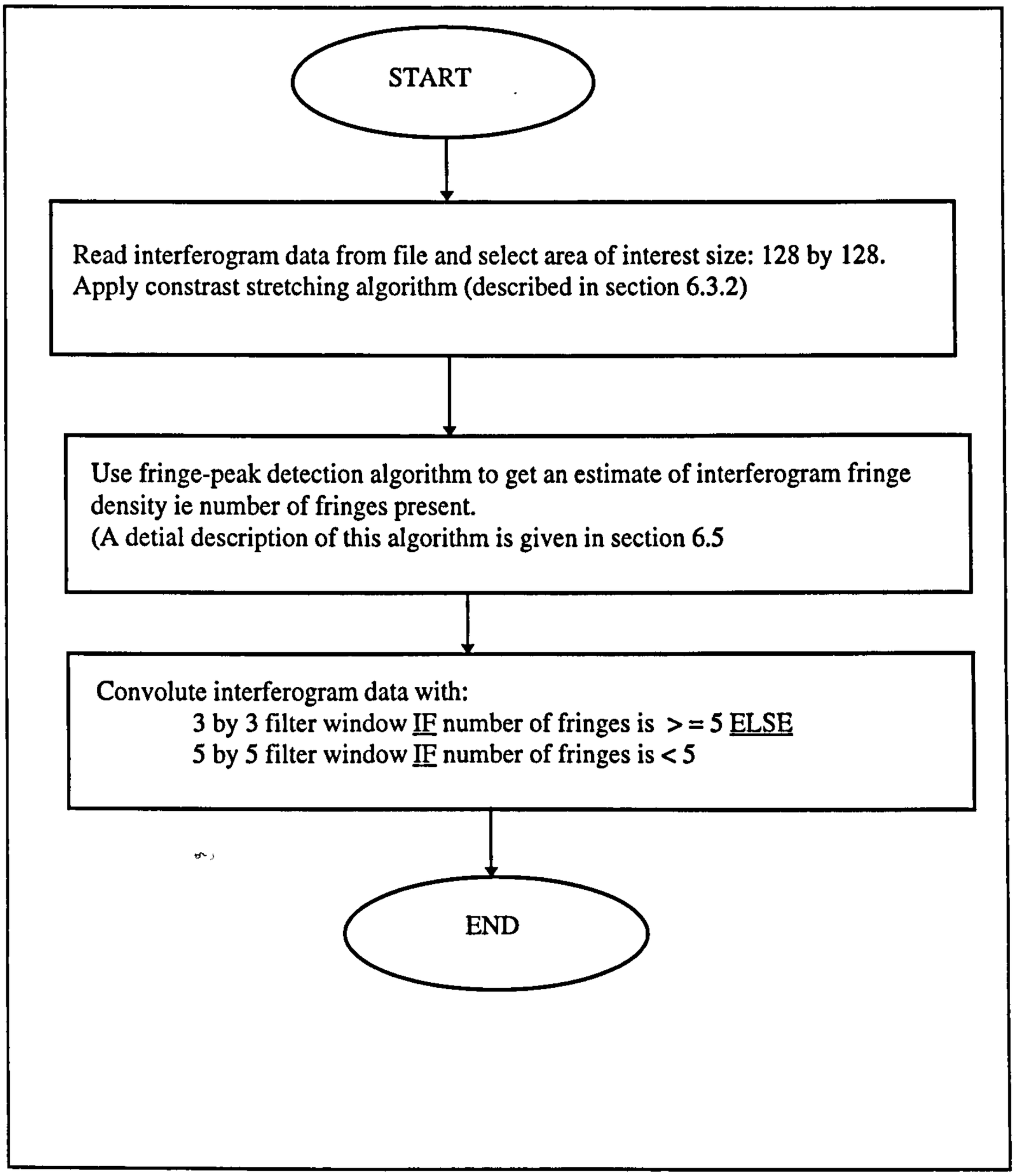
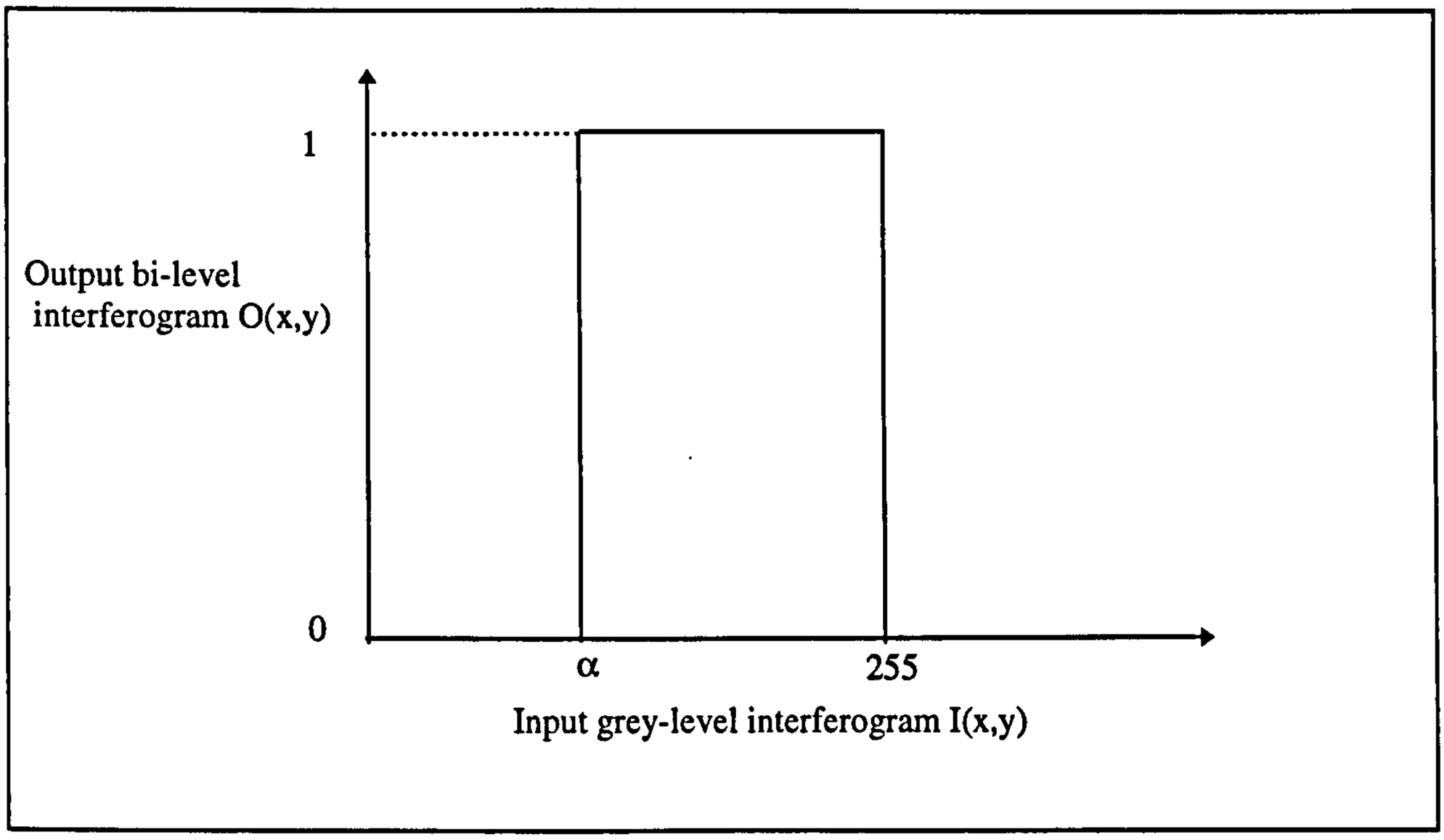


Figure 6.22: Flow chart of the spatial adaptive filtering function

The main difficulty with thresholding is finding a suitable value of α. In the particular case when the grey-level histogram of the interferogram is bi-modal (ie meaning it has two distinctive peaks as shown in figure 6.21b), α may be set to the value between the peaks (Parker, 1994). Unfortunately, analysis of various interferograms during this work indicated that not all the fringe patterns exhibit bi-modal histograms.

The method used for detecting α in this work is based on a statistical algorithm developed by Otsu (1979) known as *optimal thresholding* (Sonka et al, 1995). This approach classifies the analysed interferogram into two groups, where the first group



Figures 6.23: Graphical description of thresholding

(G_1) is the background (bright fringes) and the second group (G_2) is the object (dark fringes). The technique uses a discriminate criterion to maximise the seperability between the two groups. According Otsu (1979), the equation of group seperability (S) is:

$$\sigma^2 = \frac{(\mu(T)\omega(k) - \mu(k))^2}{\omega(k)[1-\omega(k]} \qquad (6.14)$$

Where k : range of threshold values
$\sigma^2(k)$: group variance at threshold values
$\mu(T)$: total mean value of interferogram before thresholding
$\mu(k)$: mean value of interferogram
$\omega(k)$: measure of group occurrence at threshold values

The optimal threshold value of interferogram occurs when:

$$\text{MAXIMUM}[S(k)] = \text{MAXIUM}[\sigma^2(k)] \qquad (6.15)$$

which occurs when $k = k_{max} = \alpha$.

Where k_{max} is the threshold value that gives the *maximum* value of S. Figure 6.26 shows a modified version of Otsu's (1979) thresholding algorithm used in this work.

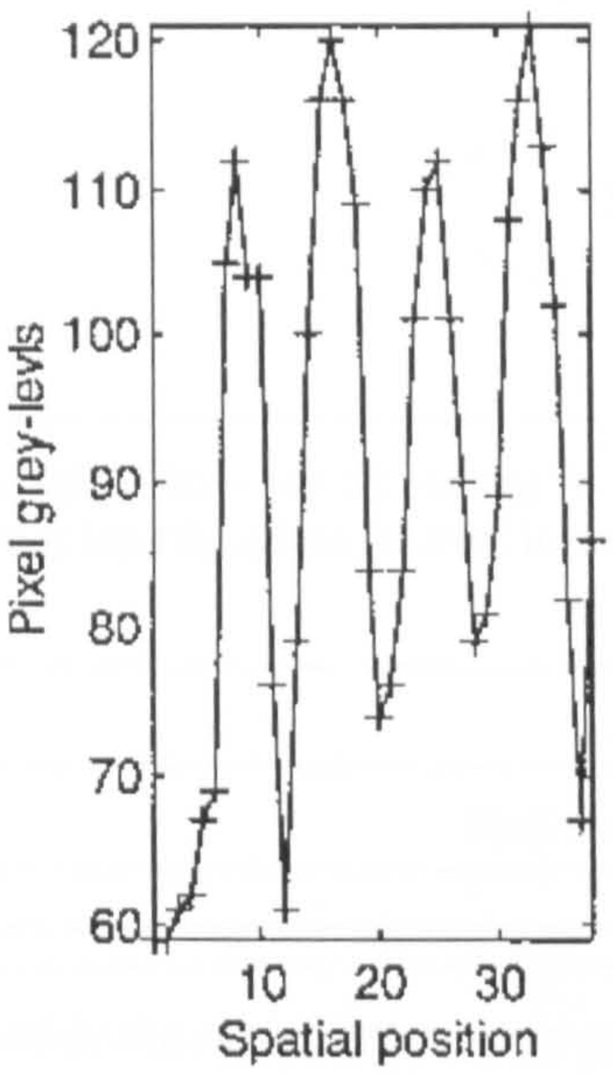


Figure 6.24a

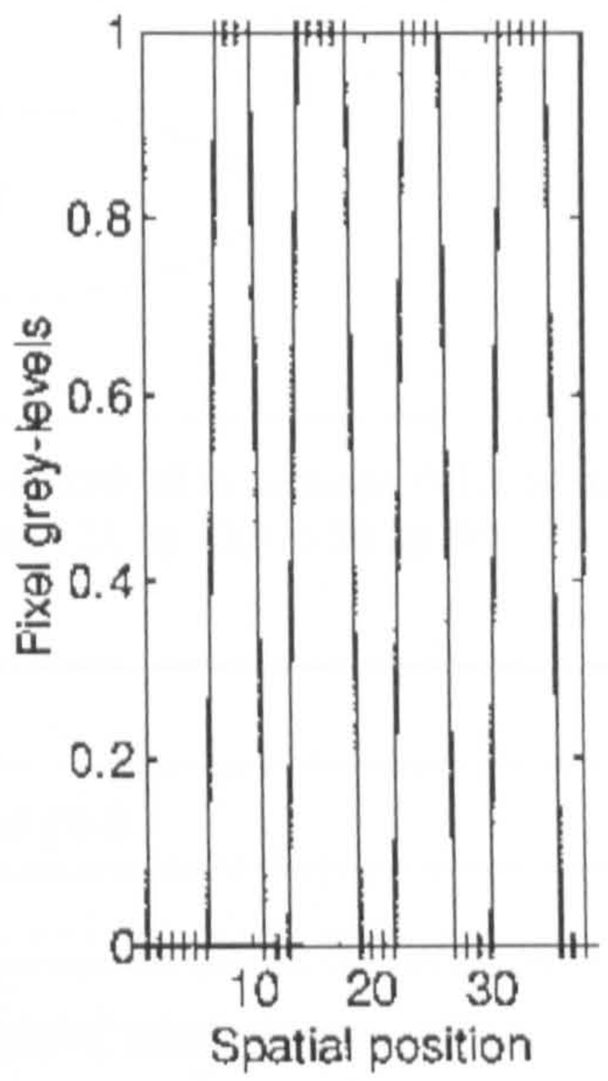
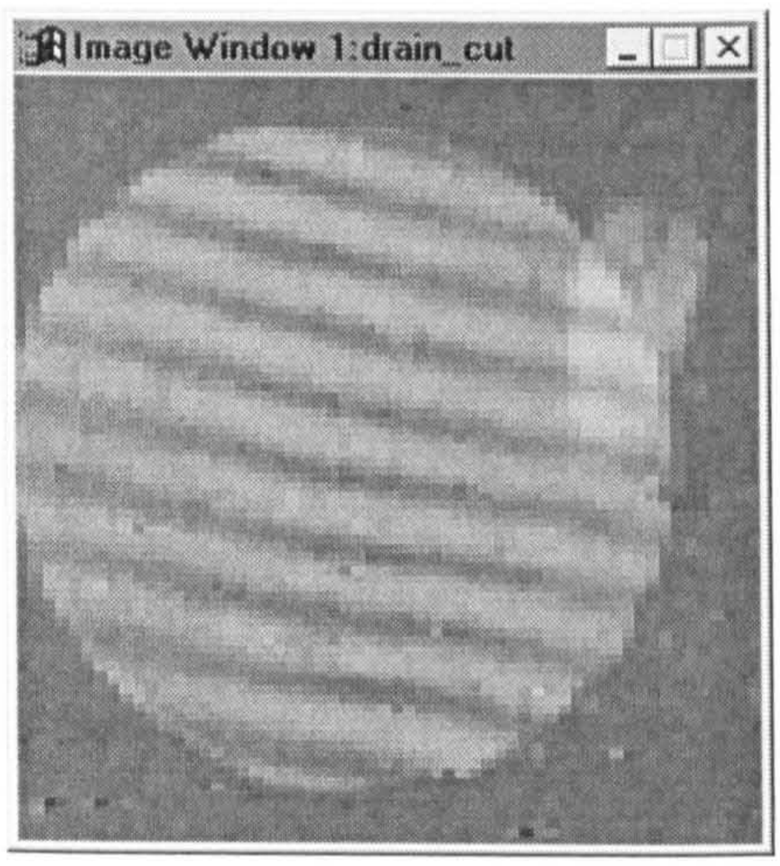


Figure 6,24b

Figure 6.24(a ,b): Intensity profile of interferogram (a) before and (a) after thresolding



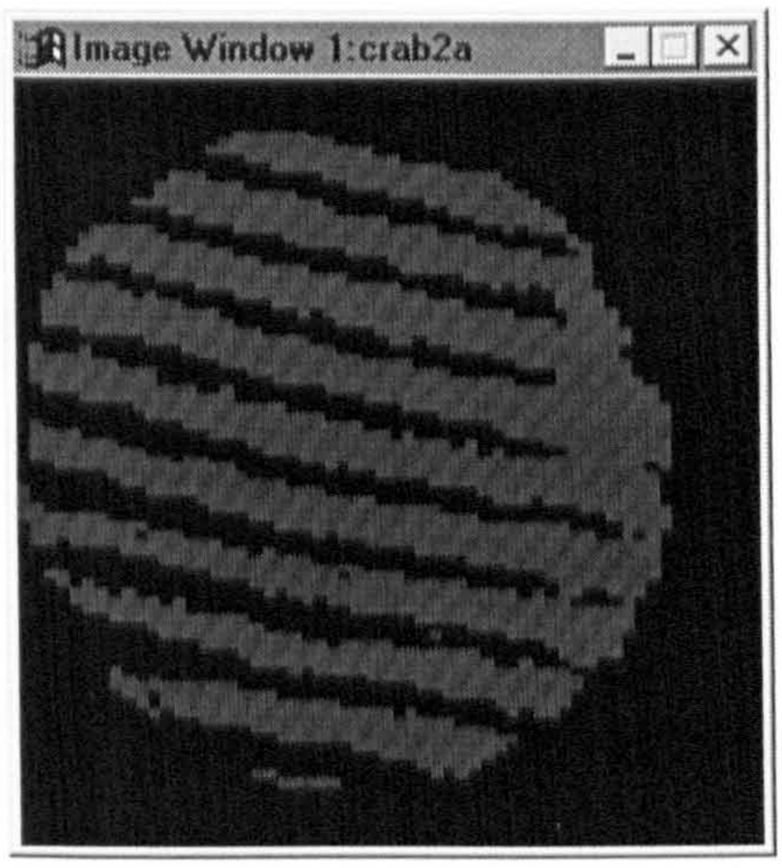


Figure 6.25a and 6.25b: Interferogram before and after thresholding

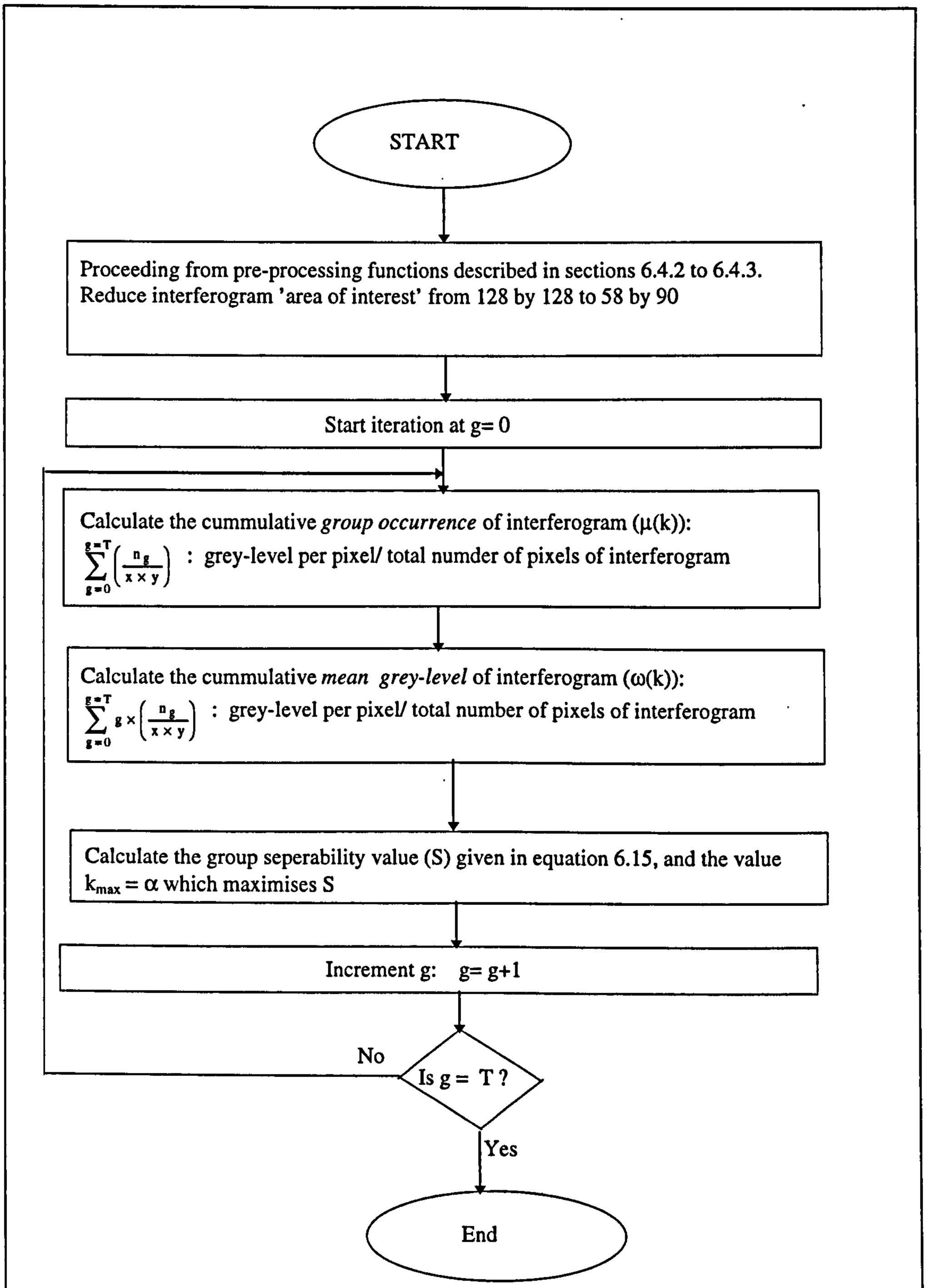


Figure 6.26: Modified version of Otsu's (1979) thresholding algorithm used in this work.

6.4.6 SPATIAL FREQUENCY DOMAIN FUNCTIONS

6.4.6.1 INTRODUCTION

There are a number of the benefits in processing an image in its Fourier domain. These include:

- The ability to detect specific geometric features such as periodicity, orientation and spacing (Russ, 1993).

- Detection and filtering of image spatial frequencies (such as its DC content, and its high/low frequencies) is easier to perform (Russ 1993, Visilog, 1993).

- Some classical image processing functions usually performed in the spatial domain are much more computational efficient when performed in the frequency domain (Visilog, 1993).

In this work a fast Fourier transform (FFT) algorithm was used to remove the DC content of interferogram and to detect its orientation. A typical interferogram (I(x,y)) can be transformed into its Fourier domain (F(u,v)) using the following equation (Chapara & Canale, 1990):

$$F(u,v) = \sum_{x=0}^{x=N-1} \sum_{y=0}^{y=N-1} \left(I(x,y) e^{-2\pi i(ux+vy)/N} \right) \qquad (6.14)$$

where x, y : rows and columns of the interferogram in its spatial domain
u, v : rows and columns of the interferogram in its frequency domain
N : Total number of row or (column) pixels in the spatial domain.

The Fourier transform is usually represented in its *real* and *imaginary* form:

$$F(u,v) = F_R(u,v) + iF_I(u,v) \qquad (6.15)$$

where the modulus of F(u,v) is:

$$\sqrt{\left(F_R^2 + F_I^2\right)} \qquad (6.16)$$

its power spectrum (P(u,v)) is:

$$\left(F_R^2 + F_I^2\right) \qquad (6.17)$$

and its phase ($\varphi(u,v)$) is:

$$\tan^{-1}\left(\frac{F_R}{F_I}\right) \qquad (6.18)$$

The fast Fourier transform (FFT) of an interferogram (figure 6.27a), sampled during rotation of the spindle is shown in figure 6.27b. Figure 6.27b indicates the orientation and sinusoidal nature of the interferogram. It also shows the DC content (ie zero harmonic peak) located at the centre (ie at u = 0 and v = 0) and the corresponding low frequency noise components surrounding it. It also shows the point spread functions (ie first harmonic peaks) of the interferogram located at positions equidistant from the zero harmonic peak.

6.4.6.2 FREQUENCY DOMAIN FILTERING

A complementary approach to interferogram spatial filtering (described in section 6.4.2) is frequency domain filtering (Low 1991, Russ 1993, Marion, 1991). This involves transforming the fringe pattern from its spatial to frequency domains using a FFT algorithm, and filtering spatial frequencies not related to the sinusoidal function of the interferogram. This technique requires some a priori knowledge of the spatial frequencies (ie DC, low or high) present in the noisy interferogram signal.

A window of dimensions 16 by 16 (positioned at co-ordinates u = 57 and v =59) was extracted from the original FFT image. This window corresponded to the area occupied by the interferogram zero harmonic and other low frequency components, it was used to act as a high pass filter as illustrated in figures 6.27(a) to 6.27(f). The resultant FFT image and interferogram due to the extraction of the DC and low frequency noise is shown in figure 6.27(f). The noise levels in the interferogram shown in figure 6.27(e) are much lower than those in original pattern shown in figure 6.27(a), making the process of detecting fringe maxima less cumbersome.

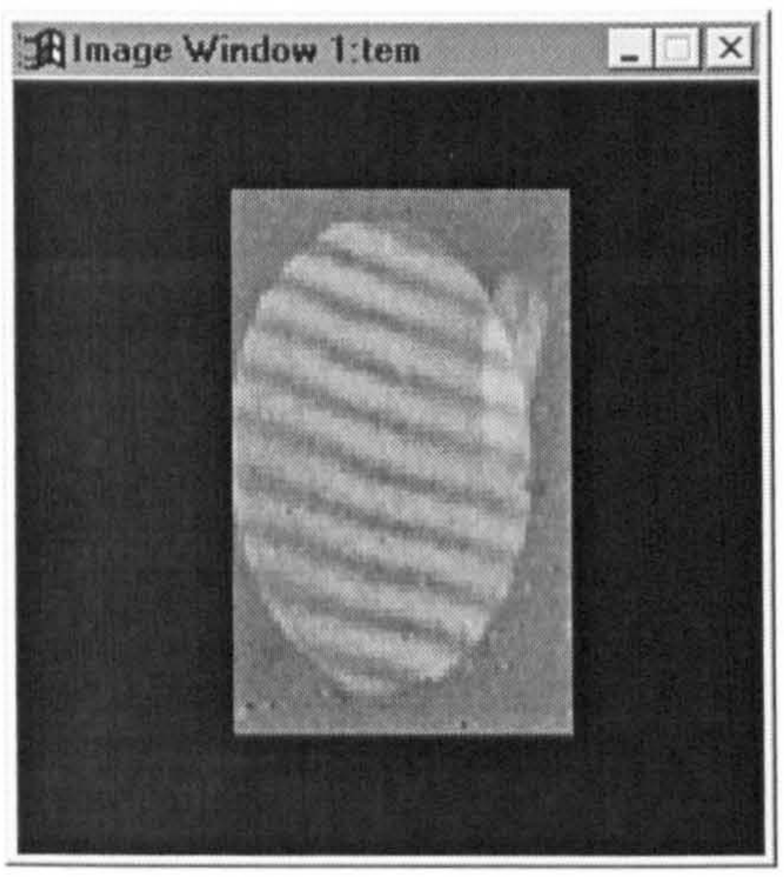
The flow-chart shown in figure 6.28 gives a detailed description of the FFT routine developed in this project.

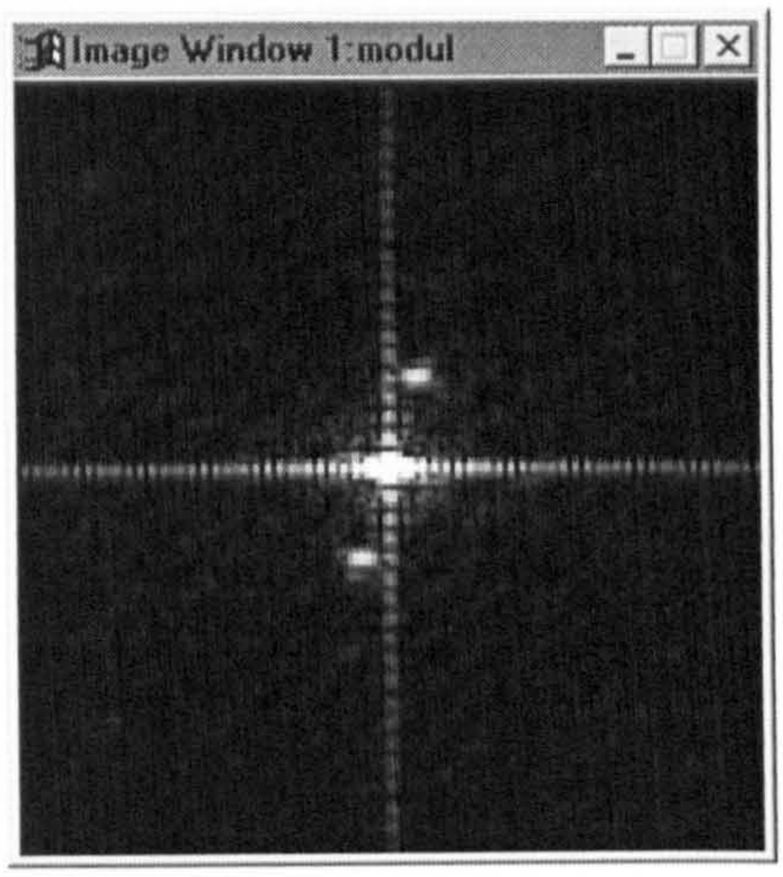
6.4.6.3 MEASUREMENT OF FRINGE ORIENTATION

In this work the interferogram orientation was evaluated from its point spread function (ie from its fundamental and first harmonic peaks present in the frequency domain). Calculation of the fringe pattern orientation requires an accurate measurement of interferogram FFT harmonic peaks.

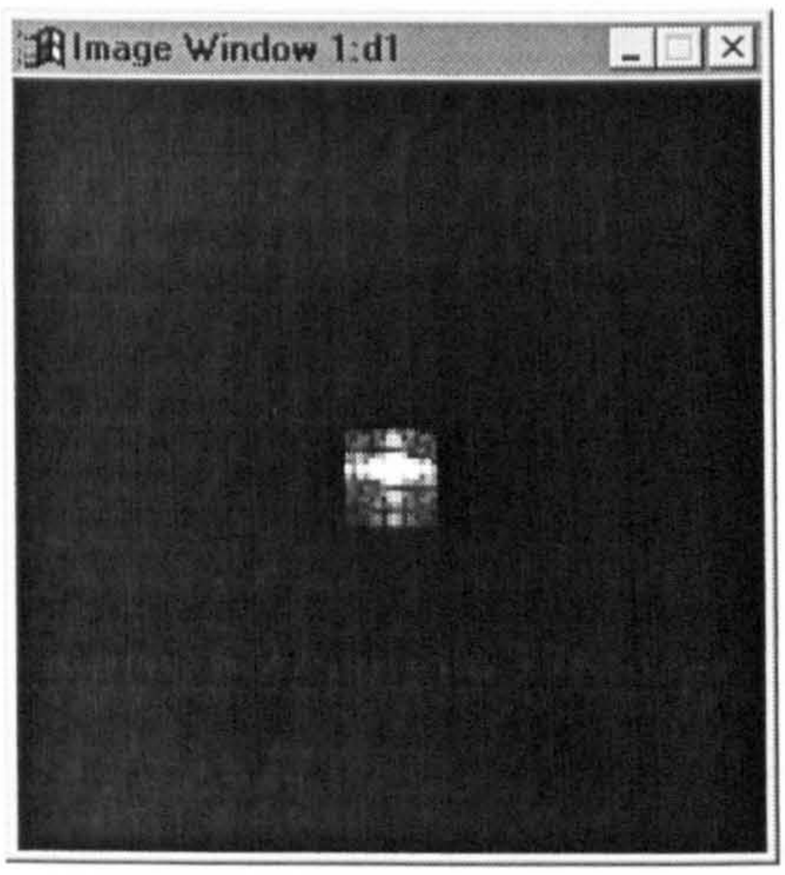
Measurement of interferogram harmonic peaks involved the following stages:

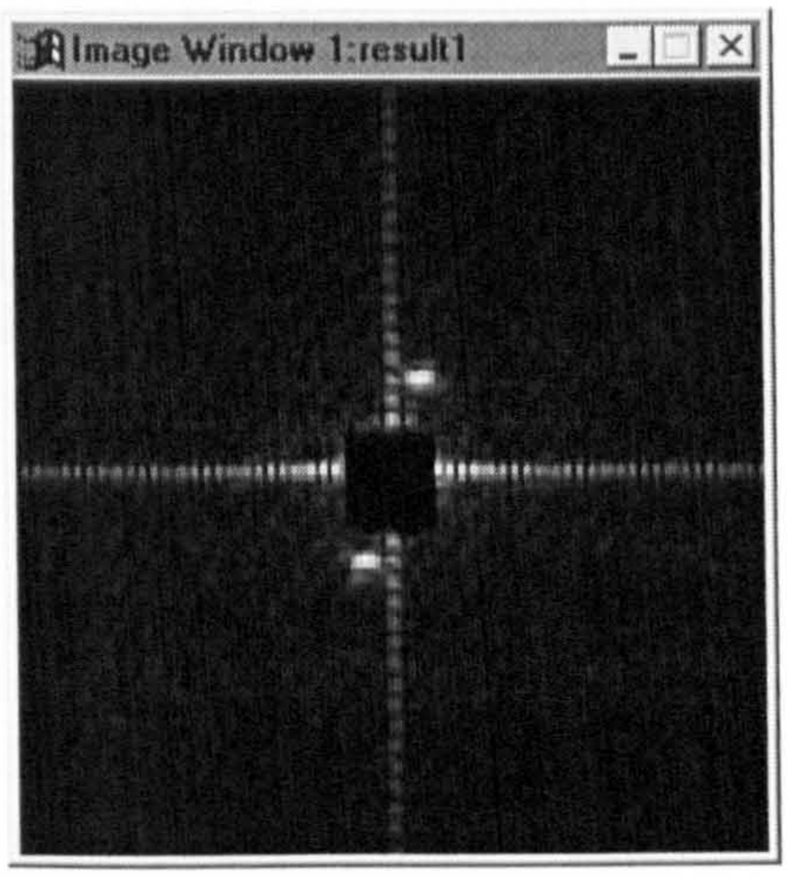
- Pre-thresholding functions
- Hysteresis thresholding
- Post thresholding
- Harmonic peak detection and labelling
- Measurement of interferogram orientation



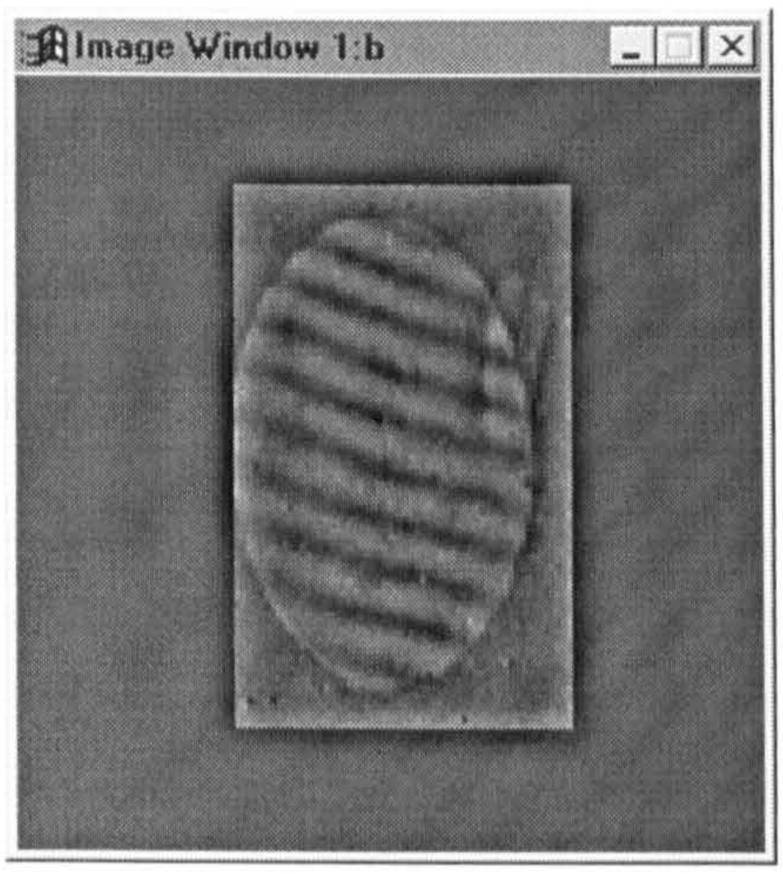


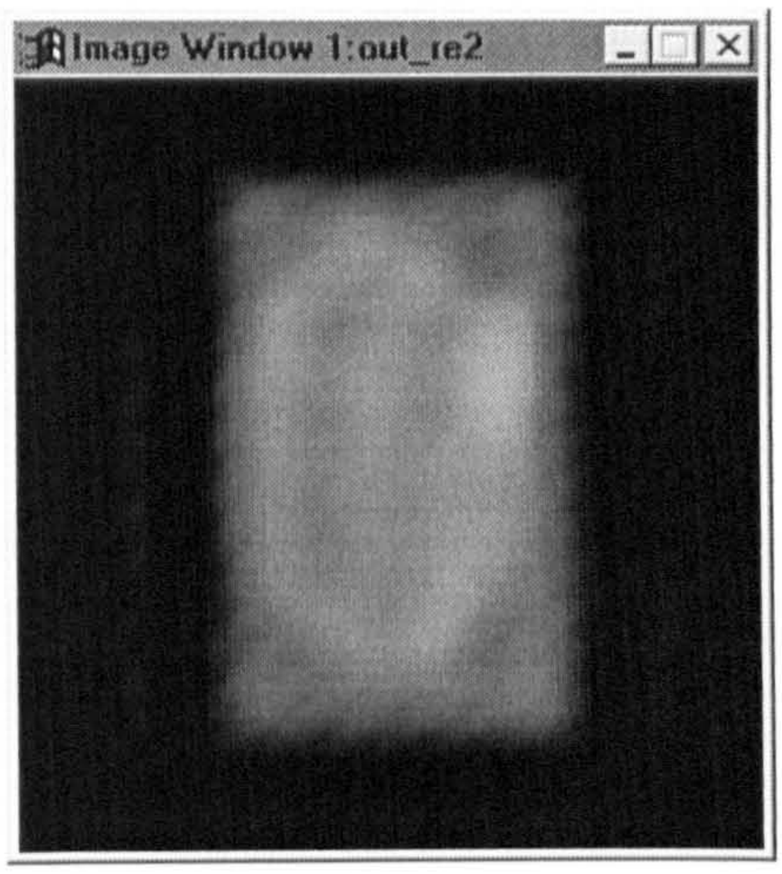
Figures 6.27a and 6.27b: Interferogram in spatial and Fourier domain





Figures 6.27c and 6.27d: DC and high pass components after filtering





Figures 6.27e and 6.27f: DC and high-pass components of Interferogram after filtering

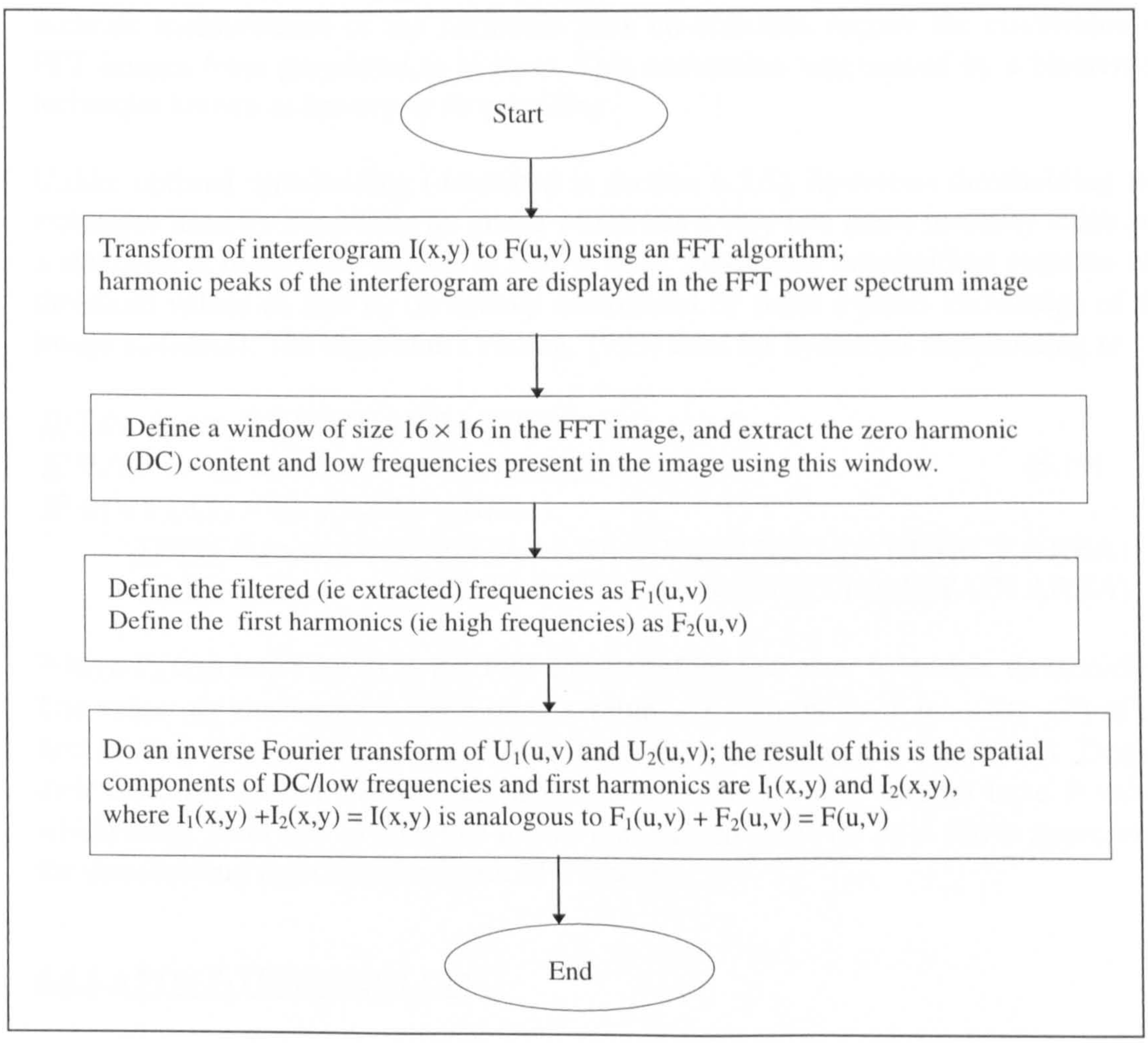


Figure 6.28: Description of frequency domain filtering using FFT technique

6.4.6.4 PRE-THRESHOLDING

The pre-thresholding stage involves modifying interferogram FFT image (ie power spectrum) in order to optimise the process of converting it from grey-level to bi-level. It is used to enhance the edges of the harmonic peaks present in the FFT image using a technique know as *gradient morphology*. Gradient morphology (Visilog, 1993) is basically the difference between the *dilation* and *erosion* of an image, (ie see section 6.3.7 for details on morphological functions).

In order to further edge-enhancement interferogram FFT harmonic peaks, a number of arithmetic steps where carried out before and after the gradient morphological transform, as illustrated in figures 6.29a and 6.29e.

6.4.6.5 HYSTERESIS THRESHOLDING

Fast Fourier transform images of most interferograms are *grey-levelled*, meaning that accurate measurement of the harmonic peak co-ordinates require the conversion of FFT images from grey-level to bi-level. This conversion was carried by a binaristion technique known as *hysteresis thresholding*.

Unlike optimal thresholding (described in section 6.3.5), hysteresis thresholding is a technique used for binarising an image which has a very low mean intensity value and a *single-peak* histogram shown in figures 6.30. Hysteresis thresholding requires two threshold values α_1 and α_2 (ie usually determined by some a priori knowledge of the image statistics). The algorithm (Visilog, 1993) used for hysteresis thresholding is:

$$
\begin{aligned}
&\underline{\text{IF}}\ F_i(u,v) < \alpha_1\ \textbf{REJECT AREA}\ \underline{\text{THEN}}\quad F_o(x,y) = 0 \\
&\underline{\text{IF}}\ F_o(u,v) > \alpha_2\ \textbf{RETAIN AREA}\ \underline{\text{THEN}}\ F_o(x,y) = 1 \qquad (6.19)\\
&\text{IF}\ \alpha_1 < F_i(u,v) > \alpha_2\ \textbf{FUZZY AREA} \\
&\qquad \underline{\text{THEN}}\ F_o(u,v) = 0\ (\underline{\text{IF}}\ \text{PIXELS ARE CONNECTED TO}\ \textbf{REJECT AREA}) \\
&\qquad \underline{\text{ELSE}}\ F_o(u,v) = 1\ (\underline{\text{IF}}\ \text{PIXELS ARE CONNECTED TO}\ \textbf{RETAIN AREA})
\end{aligned}
$$

Where $F_i(u,v)$ and $F_o(u,v)$ is the FFT image before and after hysterisis thresholding. The value α_1 was set to 0 while α_2 is a value less than the *peak intensity* (P) of the first harmonics present in sampled FFT image $F_i(u,v)$ as shown in figure 6.31. Despite their different spatial frequencies, most of the analysed FFT images have P values which range from 158 to 165, this means that a fixed value of $\alpha_2 = 160$ is appropriate for thresholding most interferogram FFT images.

6.4.6.6 POST-THRESHOLDING

The post-thresholding stage is used to simplify the binarised FFT image using the *centroid thinning* morphological function. It is used to reduce the area of the harmonic peaks (ie which resemble *blobs*) to three discrete points as shown in figures 6.32a and 6.32b.

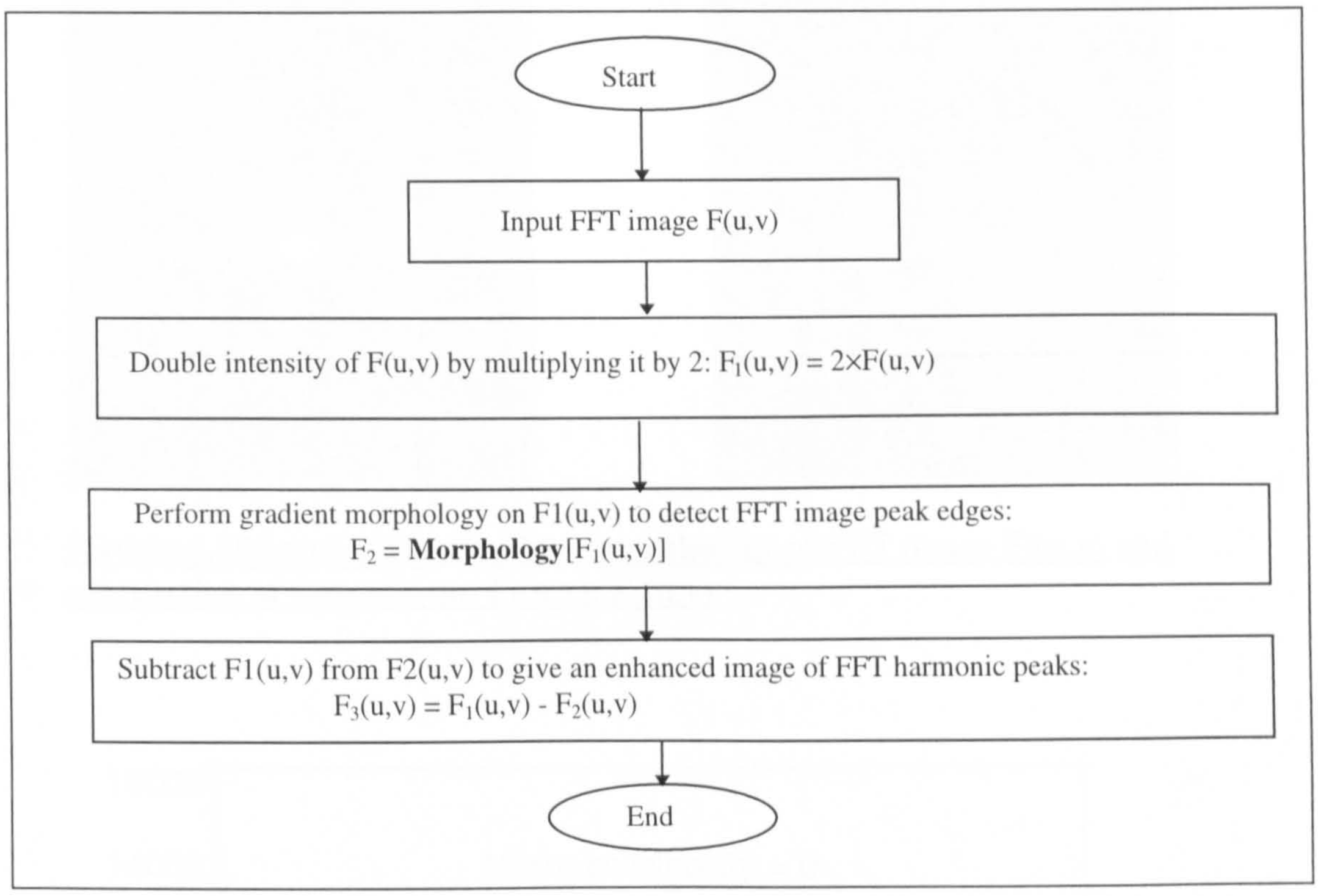
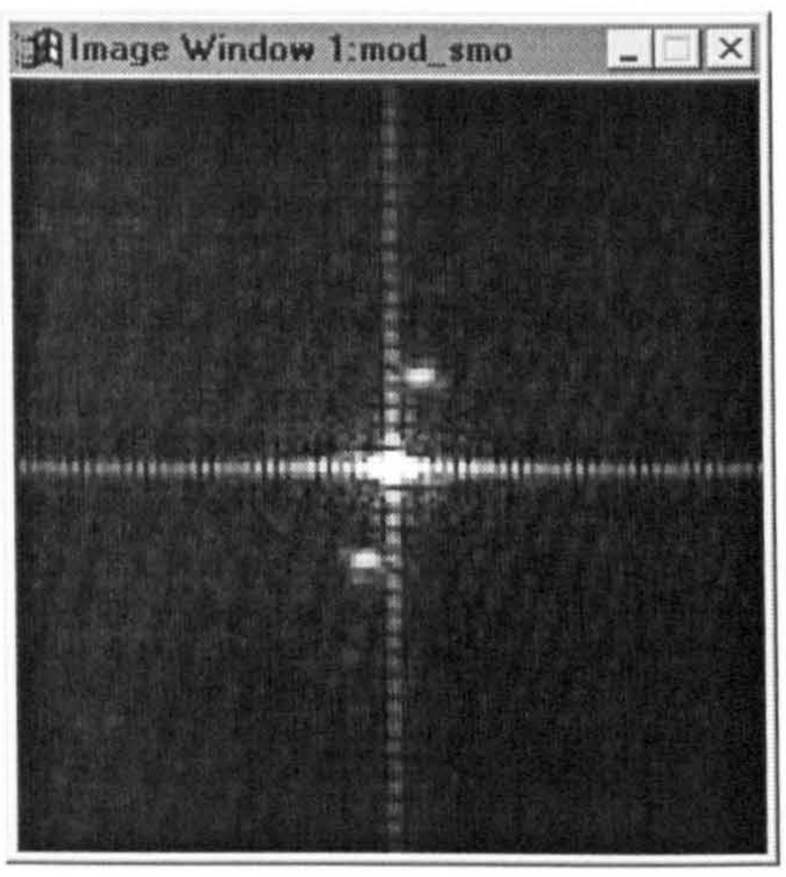
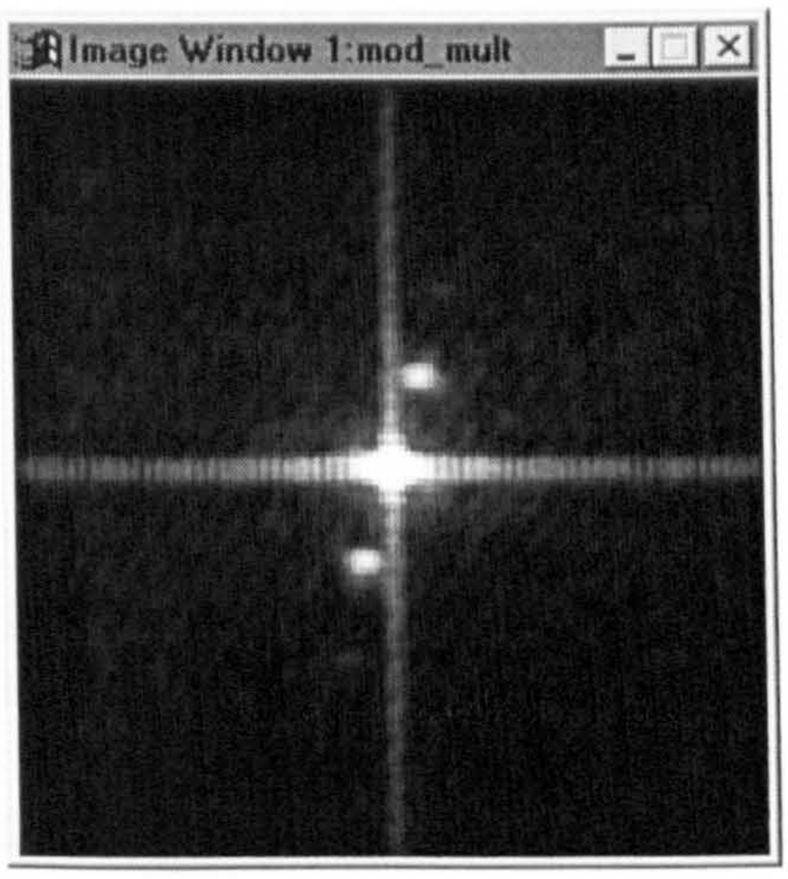
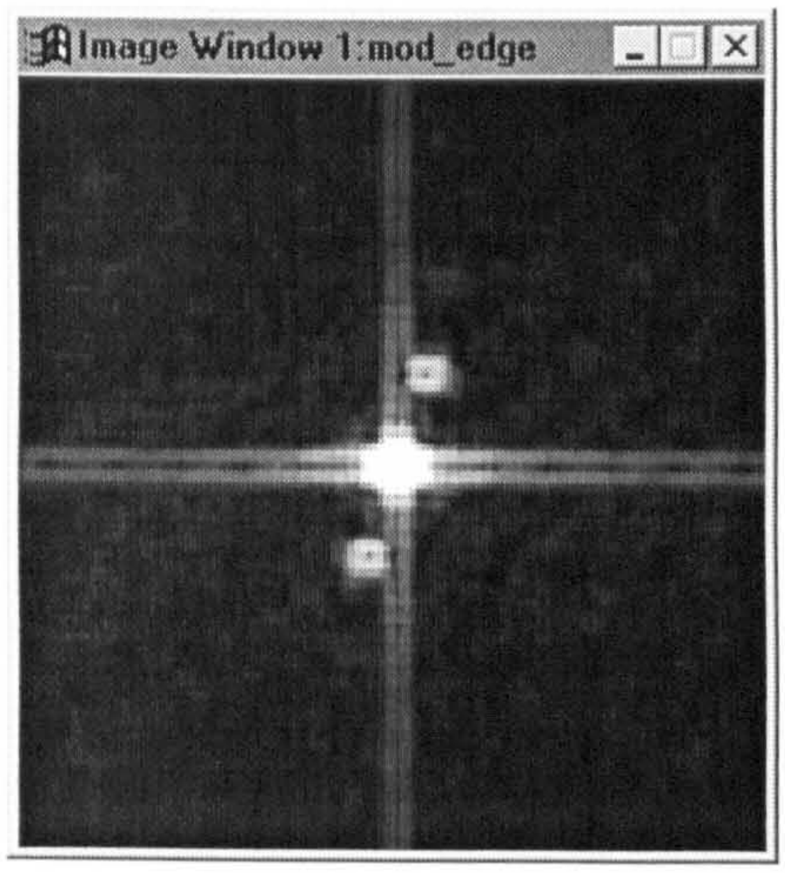


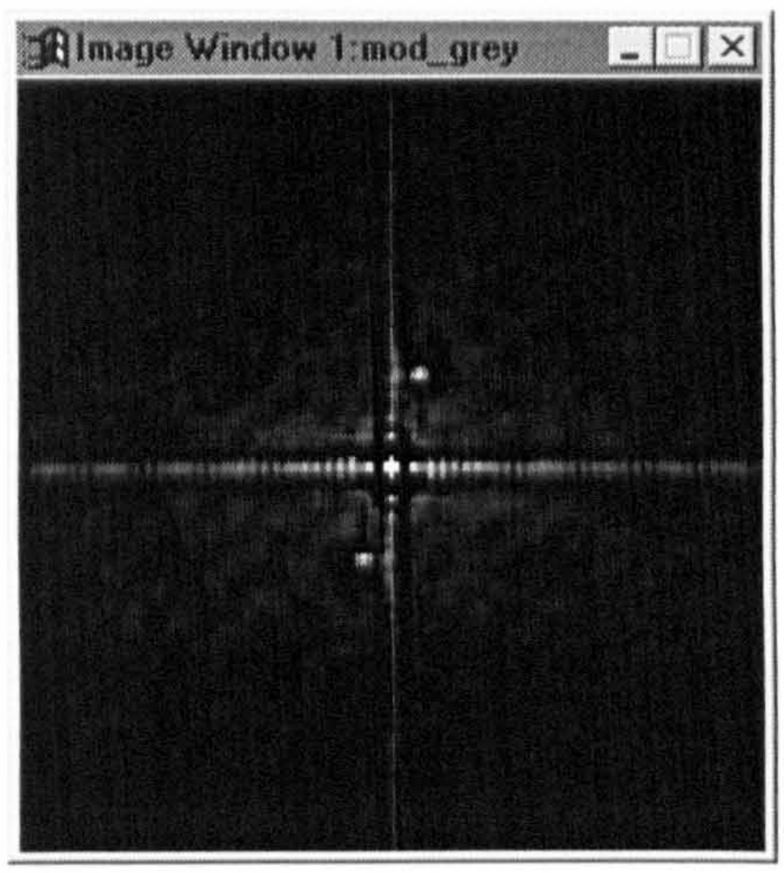
Figure 6.29a: Illustration of pre-thresholding routine





Figures 6.29b and 6.29c: FFT image of interferogram F(u,v) and FFT image with double intensity $F_1(u,v)$





Figures 6.29d and 6.29e: Gradient morphology of FFT image $F_2(u,v)$ and subtraction of $F_2(u,v)$ from $F_1(u,v)$: $F_3(u,v)$

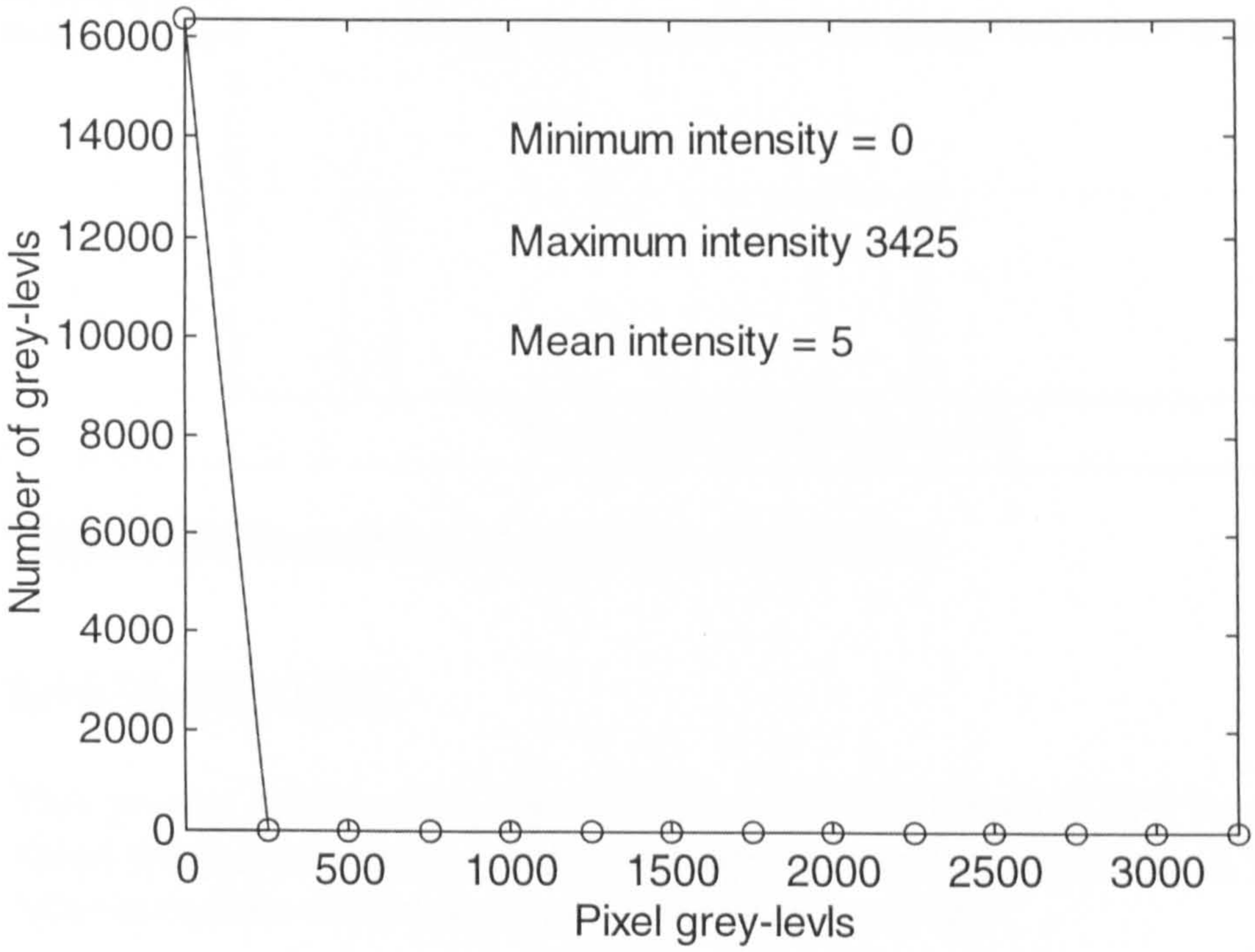


Figure 6.30: Histogram of interferogram FFT power spectrum

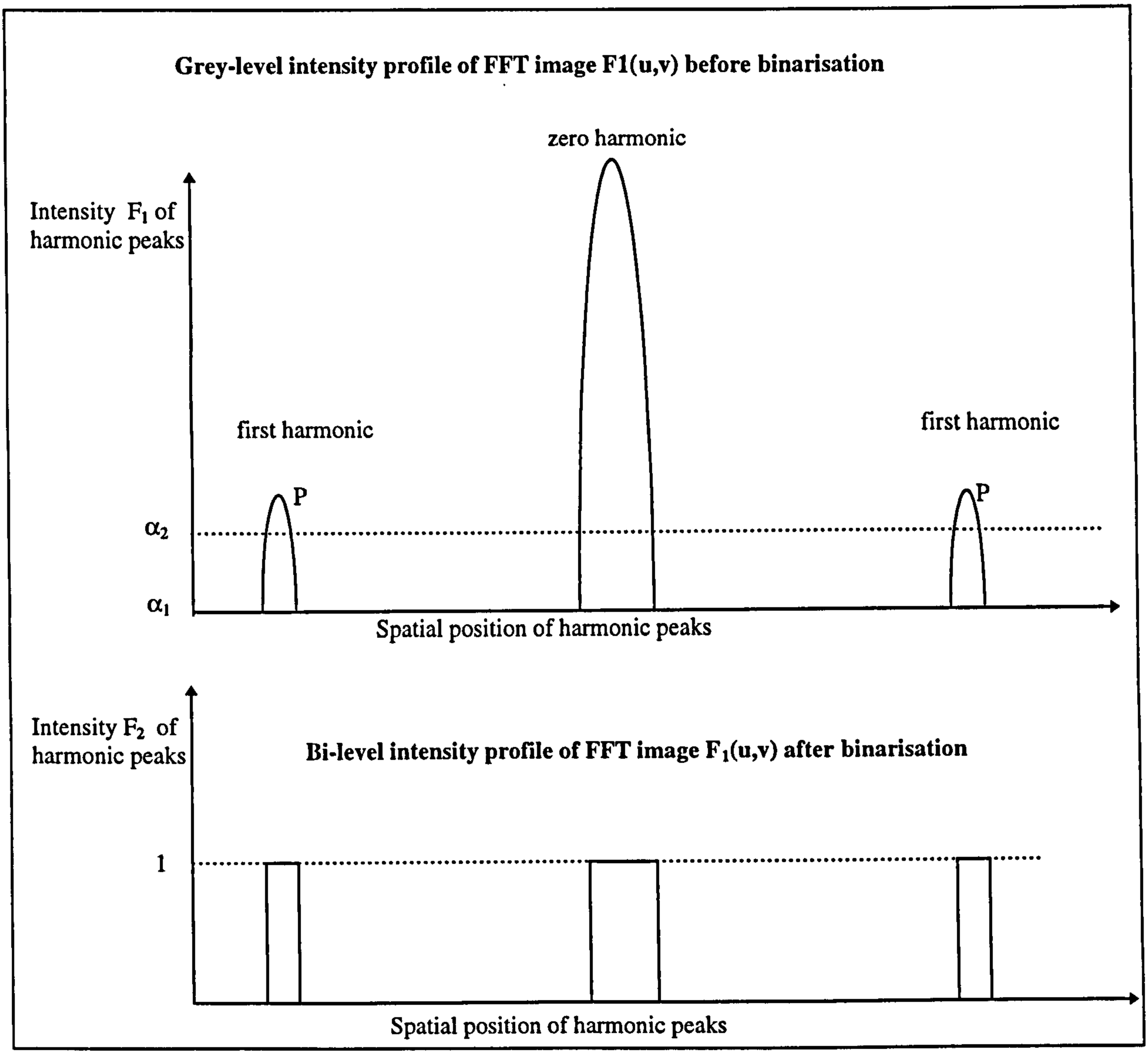


Figure 6.31: Description of hysteresis thresholding

6.4.6.7 LABELLING

This process involves the sequential scanning of a binarised FFT image in order to detect the spatial position of the three peaks present in the image. It is also used to *label* or number the harmonic peaks in a sequential order.

6.4.6.8 MEASUREMENT OF INTERFEROGRAM ORIENTATION

Interferogram orientation is based on the spatial position of the harmonic peaks and is:

$$\theta = \arctan\left(\frac{v_1 - v_2}{u_2 - u_1}\right) \tag{6.20}$$

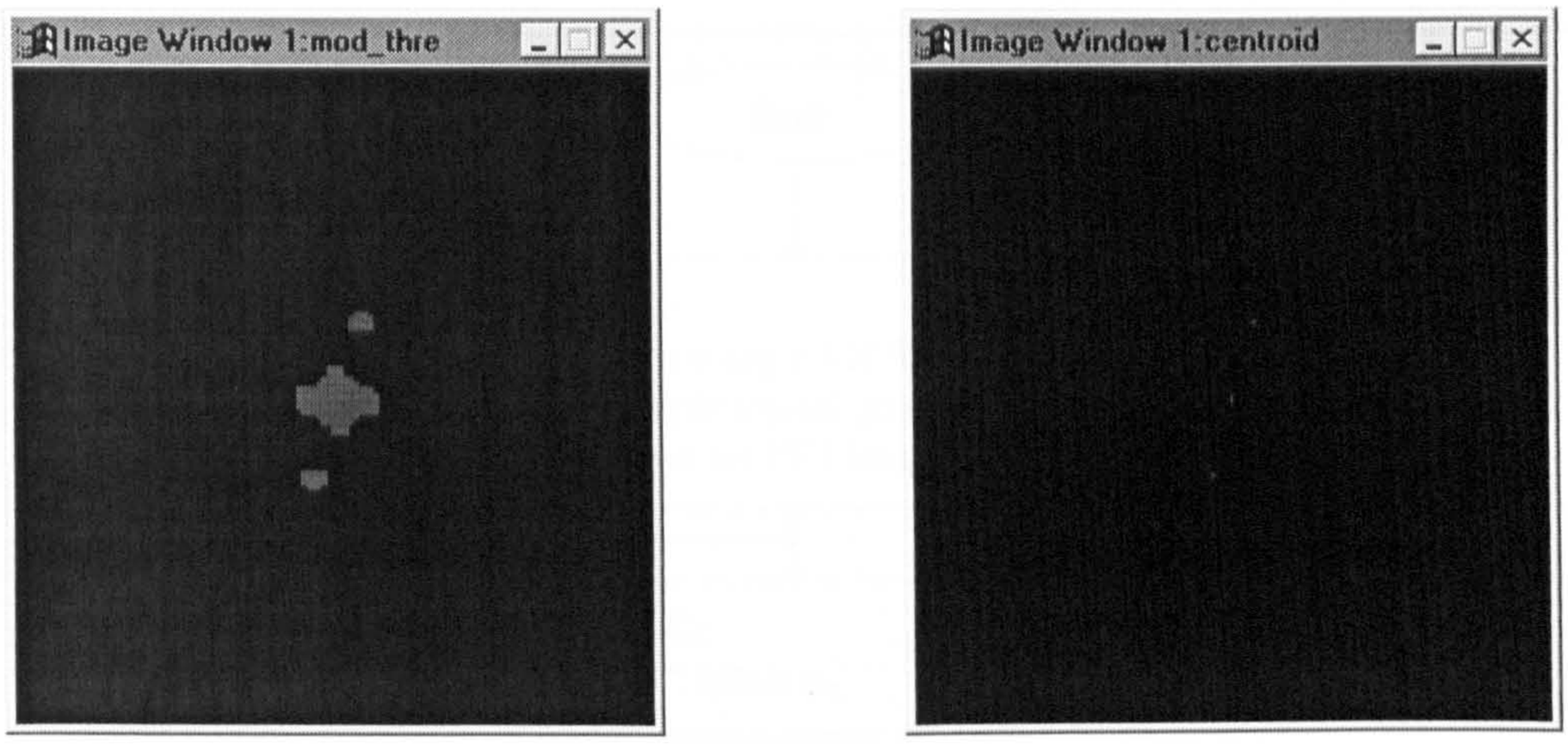


Figure 6.32a and 6.32b: Binarised FFT image before and after thresholding

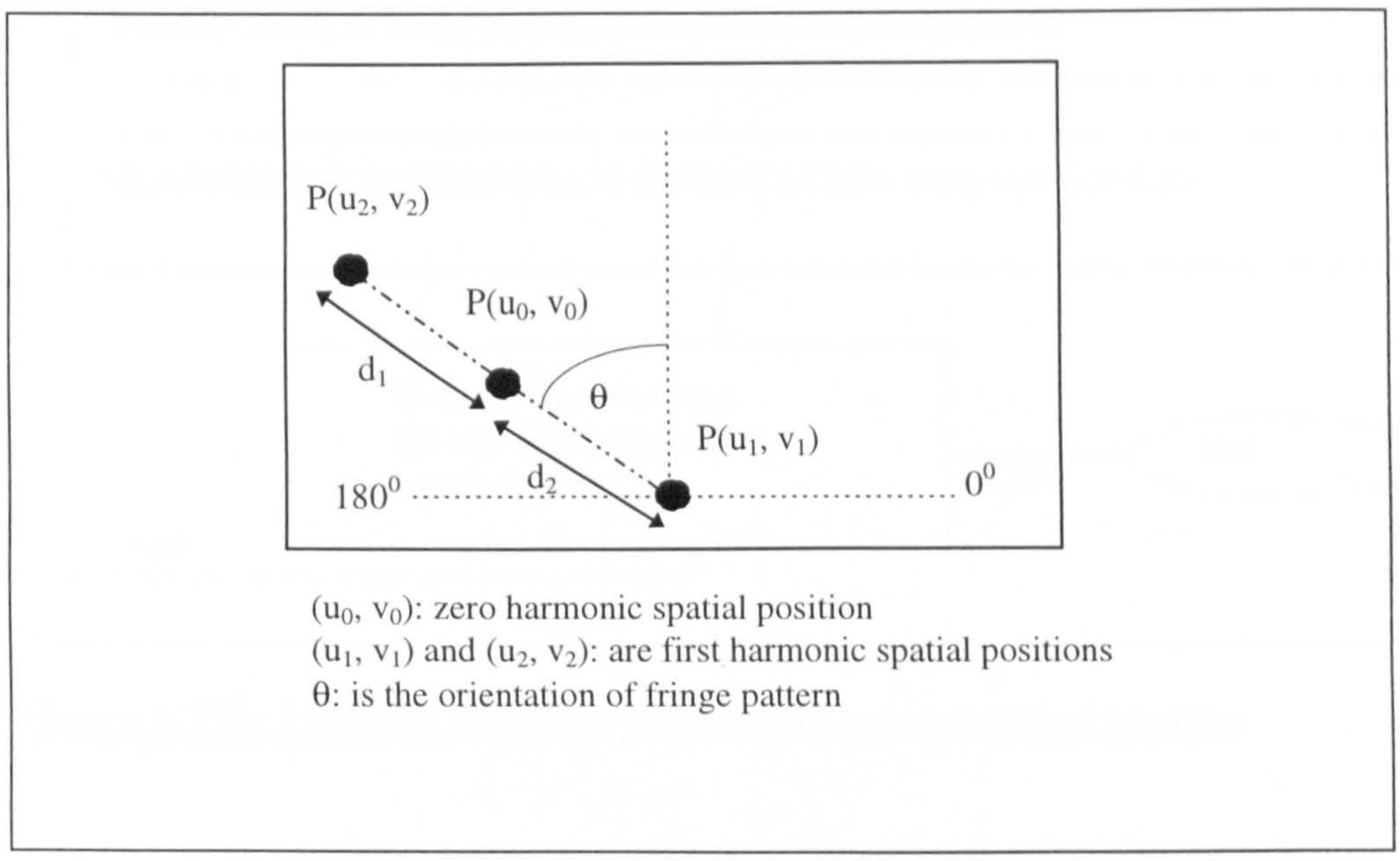


Figure 6.33a: Location of harmonic peaks of interferogram in Fourier space

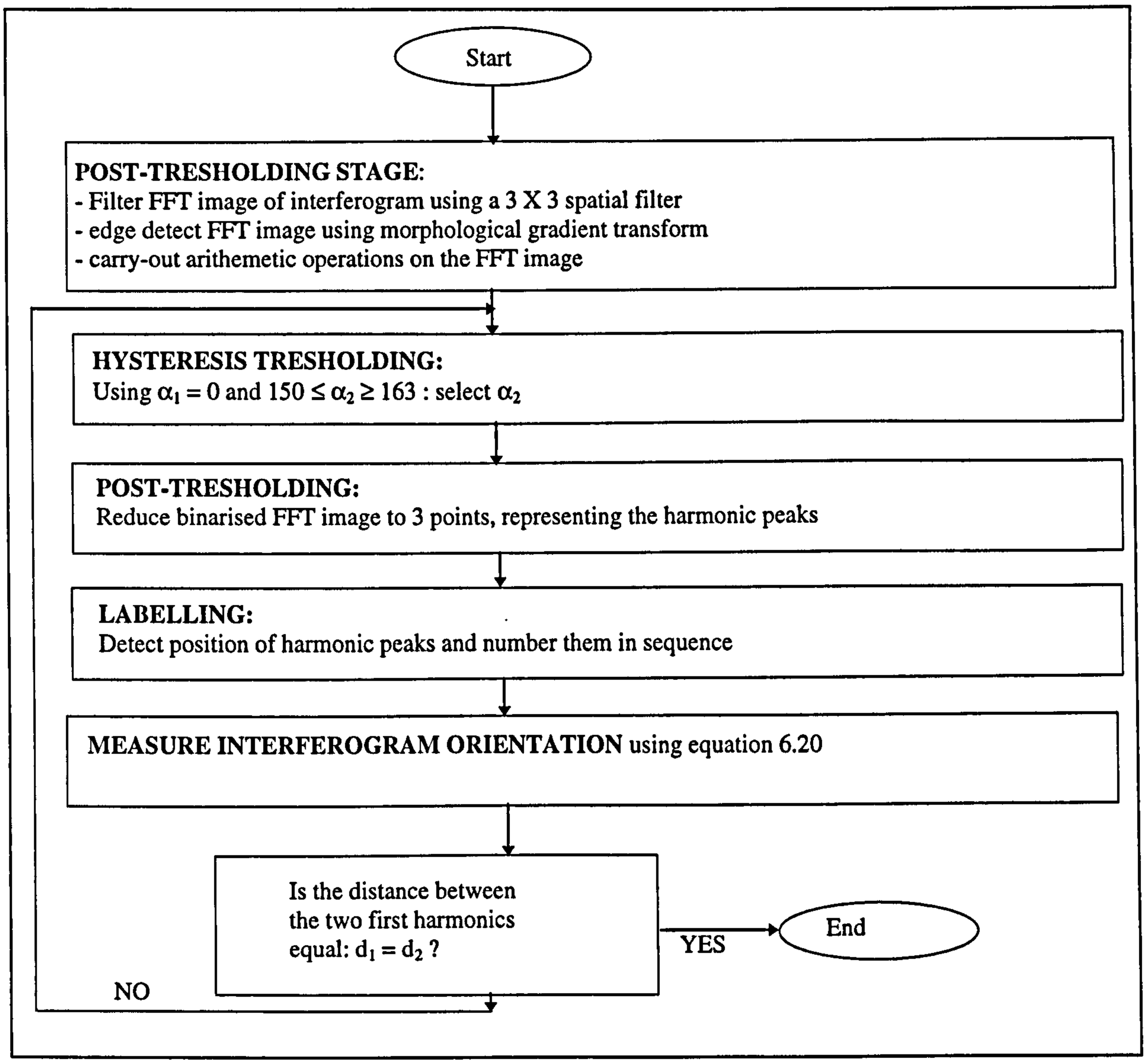


Figure 6.33b: Detailed of fringe orientation measurement routine

where u_1,v_1: spatial position of the first harmonic (positive)
u_2,v_2: spatial position of the first harmonic (negative)

The zero harmonic peak shown as $P(u_0,v_0)$ in figure 6.33a is not used to compute θ directly. However it is used to check the symmetry of the FFT image ie the distance (d_1) and (d_2) between $P(u_0,v_0)$ and first harmonic peaks $P(u_1,v_1)$ and $P(u_2,v_2)$ respectively. Ideally d_1 should be equal to d_2, any discrepancies in this requirement are checked and corrected iteratively as illustrated in figure 6.33b.

6.4.7 MORPHOLOGICAL FUNCTIONS

6.4.7.1 INTRODUCTION

A number of workers (Becker & Yu 1985, Marshall et al 1986, Hunter & Collins 1990, Yatagai 1991) involved in interferogram processing have used morphological functions (such as skeletonising, thinning, erosion etc) to simplify the geometry of fringe patterns. In this work three morphological functions were used, namely: *directional-erosion, skeletonisation*, and *centroid-thinning*.

Morphological transform of an interferogram I(x,y) requires a mobile element B called a *structuring matrix*. B is usually expressed with respect to a local origin (p) known as a *representative point* as shown in equation 6.22. During the transform, each pixel of interferogram I(x,y) is compared to B by moving the structuring matrix such that its origin *hits* all the pixels of the fringe pattern (Visilog, 1993).

Two functions that are fundamental to morphological transform of images are *erosion* and *dilation* (Awcock & Thomas, 1995), in fact the three morphological functions used in this project were derived from a combination of erosion and dilation. Erosion of an image matrix A by structuring element B, denoted as $A \ominus B$, is defined mathematically (Sonka et al, 1995) as:

$$\{p : B \subset A\} \qquad (6.21)$$

where, $\subset$: is the *inclusion* sign

An example to illustrate this function is the erosion of a hypothetical bi-level image (A) (ie it has two grey-levels of *ones* and *zeros*) by structuring matrix (B):

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \ominus \begin{bmatrix} 1^{p} & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad (6.22)$$

Input: A B Output: eroded image

On the other hand, dilation is the dual of erosion and is defined mathematically as:

$$A \oplus B = \{B: p \subset A\} \qquad (6.23)$$

An example of the dilation of a hypothetical image (A) by structuring matrix (B) is:

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \oplus \begin{bmatrix} 1^p & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} \qquad (6.24)$$

Input: A B Output: dilated image

6.4.7.2 DIRECTIONAL EROSION

Directional erosion is used to *erode* rough edges of fringes and generally smooth binarised interferograms. A unique feature of this morphological function is that the erosion of the fringes occurs in direction normal to the interferogram orientation, and this feature specifically transforms rough fringe edges to straight lines. Directional erosion of interferogram I(x,y) by a structuring matrix B is expressed mathematically (Visilog, 1993) as:

$$\overset{n=n\,max}{\underset{n=1}{}} \{ I(x,y) \Theta \; B_n^{\theta} \} \qquad (6.25)$$

where, n : is the number of iterations the interferogram is eroded

θ : orientation of interferogram after geometric transform is usually 0 degrees (ie see section 6.4.6.3 for details on measurement of fringe orientation)

$$B^{\theta} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1^p & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Figures 6.34a and 6.34b shows a binarised fringe pattern before and after directional erosion. Figure 6.34b shows the *branch-like* appearance of some of the fringes, these are usually referred to as *prunes*. Prunes can be eliminated by using skeletonisation morphology.

6.4.7.3 SKELETONISATION

Skeletonisation involves reducing the width of the fringes present in an interferogram to ideally a pixel size of unity. The skeletonisation of interferogram I(x,y) by structuring matrix B is an iterative process and is expressed mathematically (Awcock & Thomas, 1995) as:

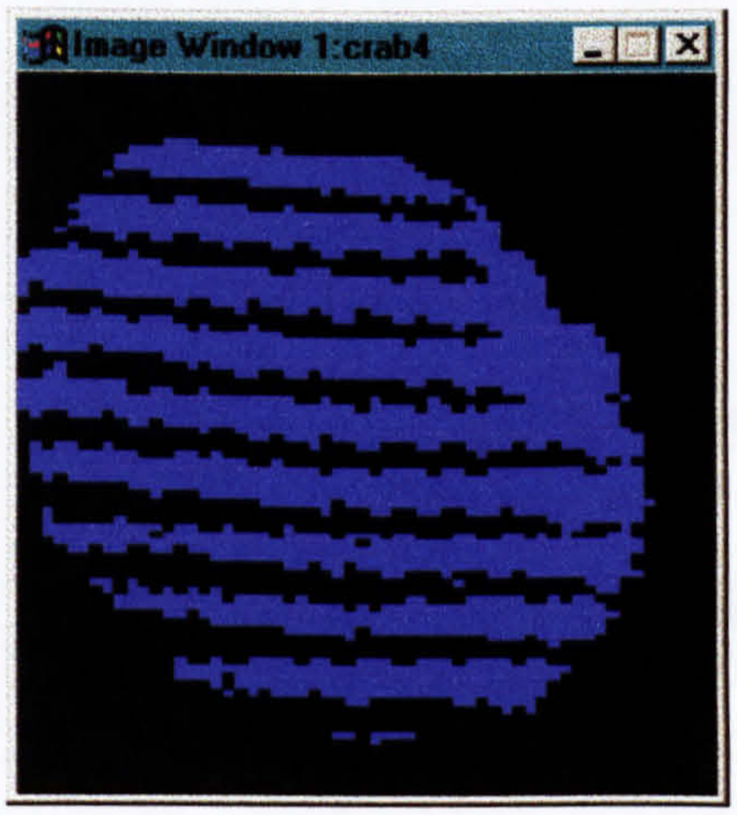


Figure 6.34a and 6.34b: Bi-level interferogram before and after directional erosion

Figure 6.35: Skeletonisation of interferogram shown in figure 6.34b

$$\bigcup_{n=0}^{n=n\,max} \left[(I(x,y)\Theta B_n) / ((I(x,y)\Theta B)_n) \oplus B_n) \right] \qquad (6.26)$$

where, $I(x,y)\Theta B_n$: indicates result of interferogram nth erosion

$((I(x,y)\Theta B_n) \oplus B_n)$: result of interferogram nth erosion is dilated n times

$/$: is the *difference* sign

$\bigcup_{n=0}^{n=n\,max}$: union of the y interferograms at n to nmax

n, n_{max} : number of iterations and maximum iteration

$$B^e = \begin{bmatrix} 1 & 1 & 1 \\ X & 1^p & X \\ 0 & 0 & 0 \end{bmatrix} \qquad (6.27)$$

where, X : is either 1 or 0 state
B^e : are the neighbourhood elements: 1, 1, 1, X, X, 0, 0 and 0

During each iteration the neighbourhood elements of the structuring matrix are rotated until convergence is reached (Sonka et al, 1995). Figures 6.34b and 6.35 shows part of a directionally eroded interferogram before and after skeletonisation respectively.

6.4.7.4 CENTROID-THINNING FUNCTION

Centroid-thinning morphology was used in this work to modify the harmonic peaks of the interferogram FFT image (ie explained in details in section 6.4.6.3.4). This function uses a structuring matrix B:

$$B = \begin{bmatrix} 0 & X & X \\ 0 & 1^p & 1 \\ 0 & 0 & X \end{bmatrix} \qquad (6.28)$$

which contains neighbouring elements: B^e = 0, X, X, 0, 1, 0, 1 and X, and is expressed mathematically (Visilog, 1993) as:

$$I(x,y) \circ B = \{0 : B^e \subset I\} \qquad (6.29)$$

Like skeletonisation, centroid-thinning morphology is usually an iterative process which involves the rotation of the neighbourhood elements of B at each iteration n. In this work the number of iterations was typically 16, this implies:

$$\bigcup_{n=1}^{n=15} \{I(x,y) \circ B_n\} = \{[I(x,y) \circ B_1] \circ B_2\} \ldots\ldots\ldots\ldots \circ B_{15} \qquad (6.30)$$

Figures 6.32a and 6.32b shows a FFT image before and after centroid-thinning.

6.5 SUB-PIXEL FRINGE PEAK DETECTION

6.5.1 INTRODUCTION

A number of sub-pixel fringe peak detection algorithms have been developed in the past (Pugh & Jackson 1982, Gasvik 1995 pp251-254, Chang et al, 1994) for intensity based interferogram analysis. In this work, an algorithm was developed to measure

fringe peak positions to an accuracy of a hundredth of a pixel. This algorithm consists of two routines:

- ESTIMATION ROUTINE: An approximation of fringe peak position of a binarised interferogram.

- INTERPOLATION ROUTINE: Maximum and minimum values computed with the estimation routine are interpolated to a polynomial form, using a least square fitting routine to give an accurate measurement of fringe maxima position.

6.5.2 ESTIMATION ROUTINE

A pre-processed interferogram (ie see section 6.3 for details) is scanned in the direction of fringe orientation to estimate the number of (bright) fringes (N) and number of *probable* maximum points per fringe (n). It is also known from a priori knowledge that each bright fringe has at least one minimum point located on each side of the fringe.

As a bi-level interferogram is scanned, the estimation routine detects and stores the spatial position of each Nth fringe maximum $(X_n)_N$, and their *left* and *right* neighbouring minimum points $(X_L)_N$ and $(X_R)_N$. The stored data in matrix form:

$$
\begin{array}{ccccccccc}
(X_L)_1 & (X_R)_1 & (X_1)_1 & (X_1)_2 & (X_1)_3 & . & . & . & (X_1)_N \\
(X_L)_2 & (X_R)_2 & (X_2)_1 & (X_2)_2 & (X_2)_3 & . & . & . & (X_2)_N \\
(X_L)_3 & (X_R)_3 & (X_3)_1 & (X_3)_2 & (X_3)_3 & . & . & . & (X_3)_N \\
. & . & . & . & . & . & . & . & . \\
. & . & . & . & . & . & . & . & . \\
(X_L)_N & (X_R)_N & (X_n)_1 & (X_n)_2 & (X_n)_3 & . & . & . & (X_n)_N
\end{array}
\qquad (6.31)
$$

is used to detect the corresponding intensity values: $I[(X_n)_N]$, $I[(X_L)_N]$ and $I[(X_R)_N]$ of the interferogram in its grey-level form, as described in figures 5.36a and 5.36b.

Figures 6.37a and 6.37b show a typical grey-level and bi-level interferogram respectively (spatial resolution of these patterns have been increased by approximately 12 times, using the pixel re-mapping technique described in section 8.3.2). Their one-dimensional intensity profiles when scanned in the direction of fringe orientation are shown in figures 6.37c and 6.37d respectively. The binarised form of the interferogram is essentially an approximation of its grey-level equivalent, and hence can be used to estimate fringe *probable* maximum positions. For instance the third fringe in figure 6.37b has $n > 50$ *probable* maximum points as shown in figure 6.37e.

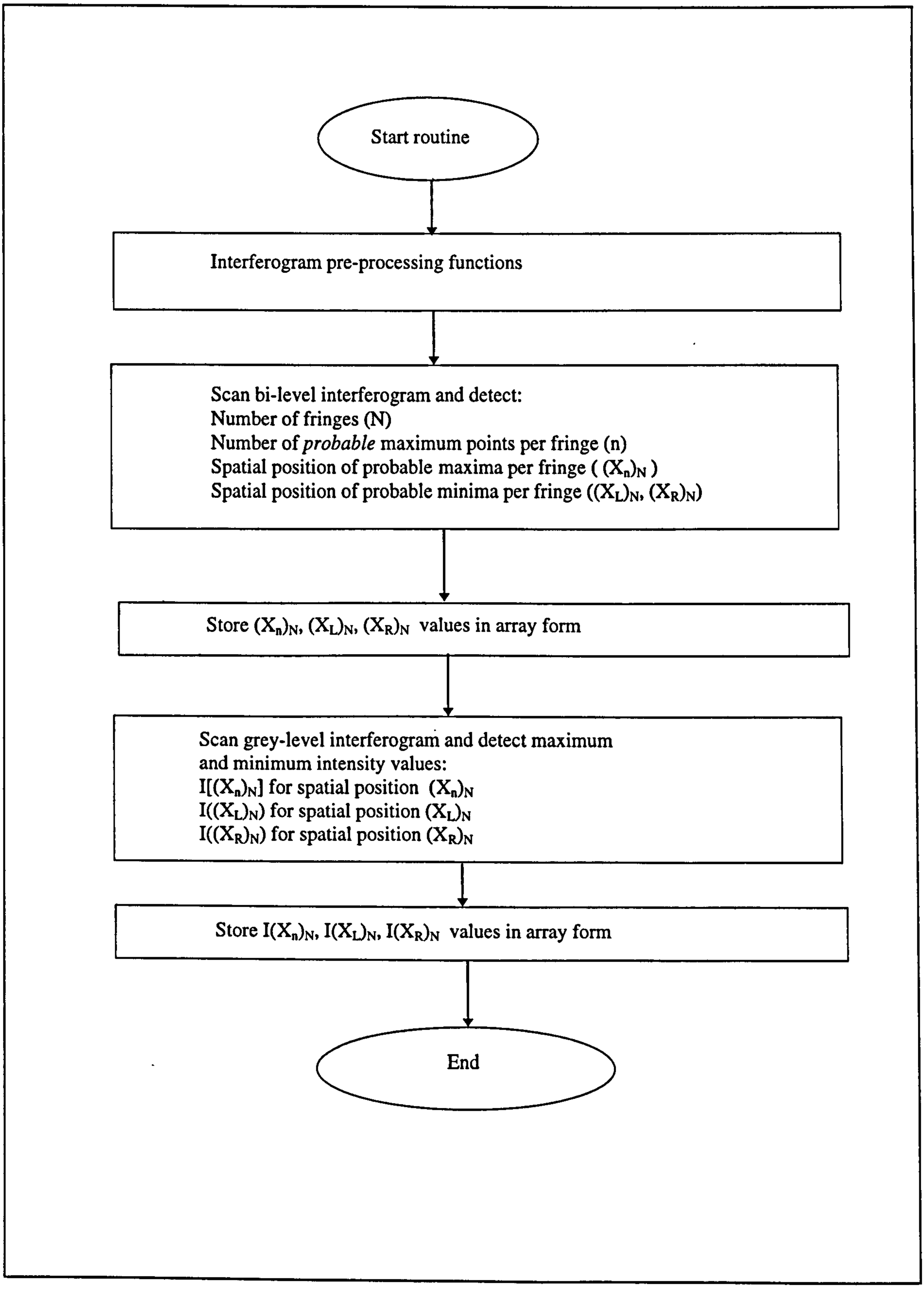
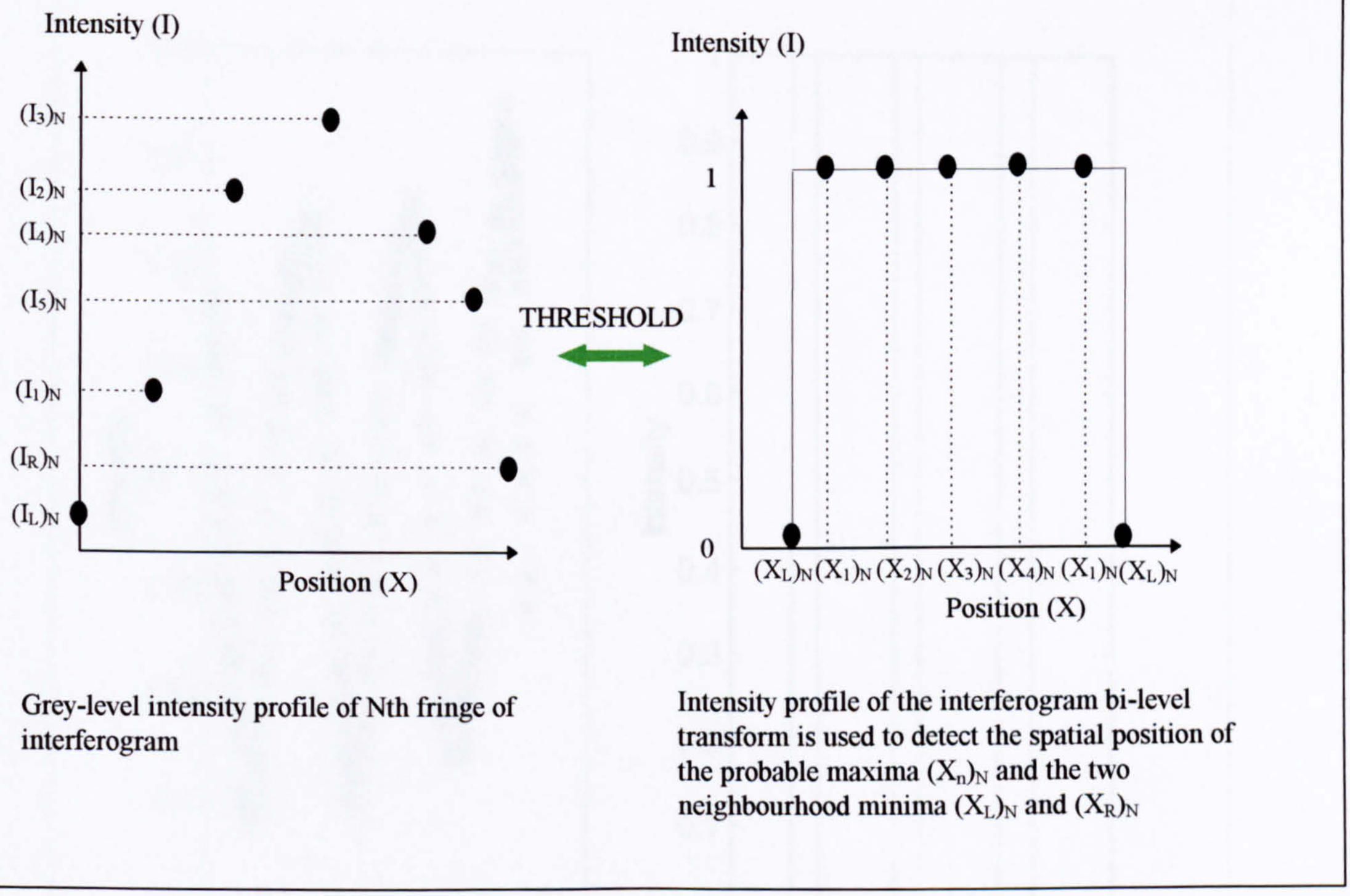


Figure 6.36a: ***Estimation routine*** **flow diagram**



Grey-level intensity profile of Nth fringe of interferogram

Intensity profile of the interferogram bi-level transform is used to detect the spatial position of the probable maxima $(X_n)_N$ and the two neighbourhood minima $(X_L)_N$ and $(X_R)_N$

Figure 6.36b: Estimation of *probable* fringe maxima

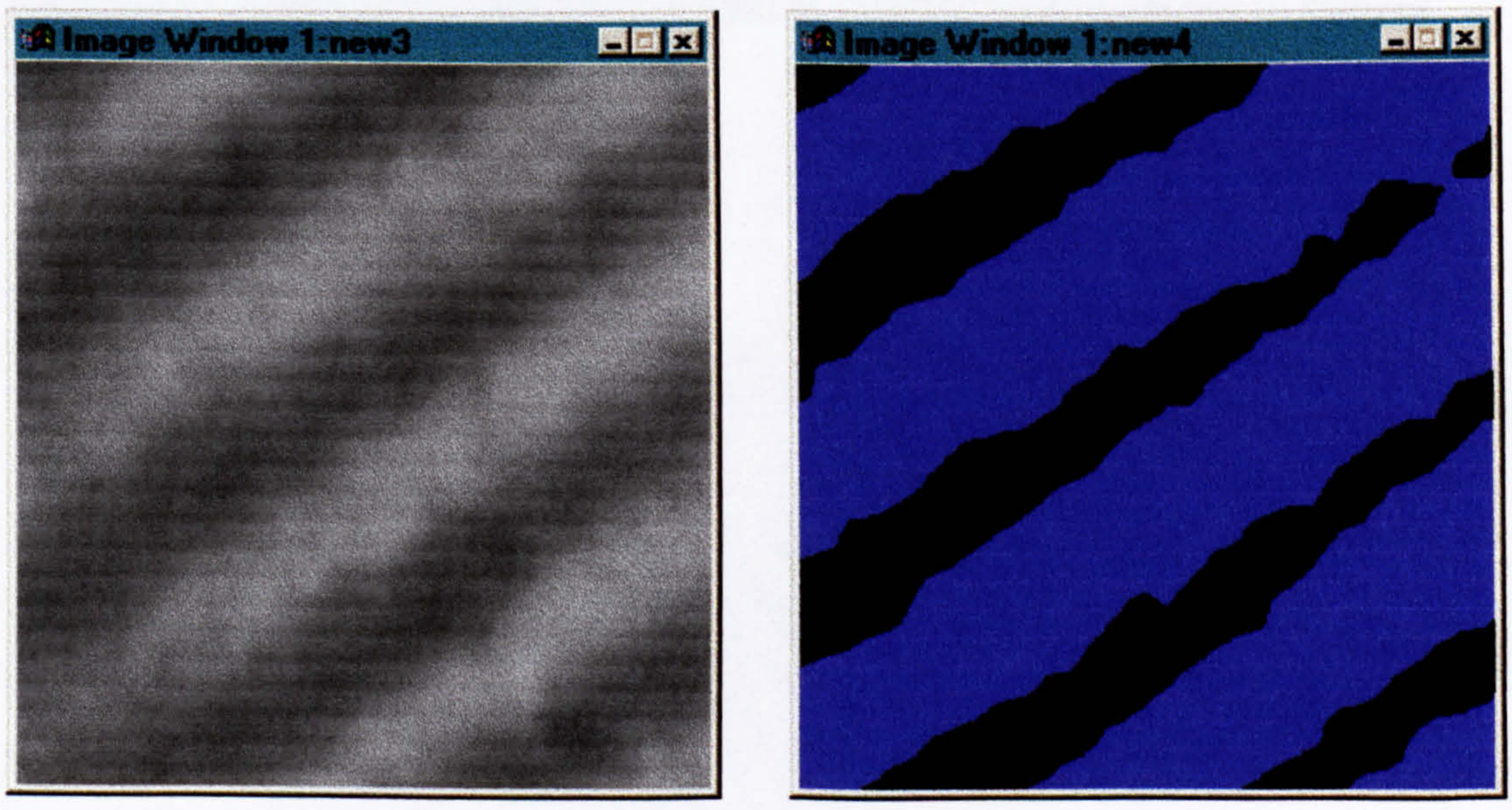


Figure 6.37a and 6.37b: Grey-level and bi-level form of interferogram

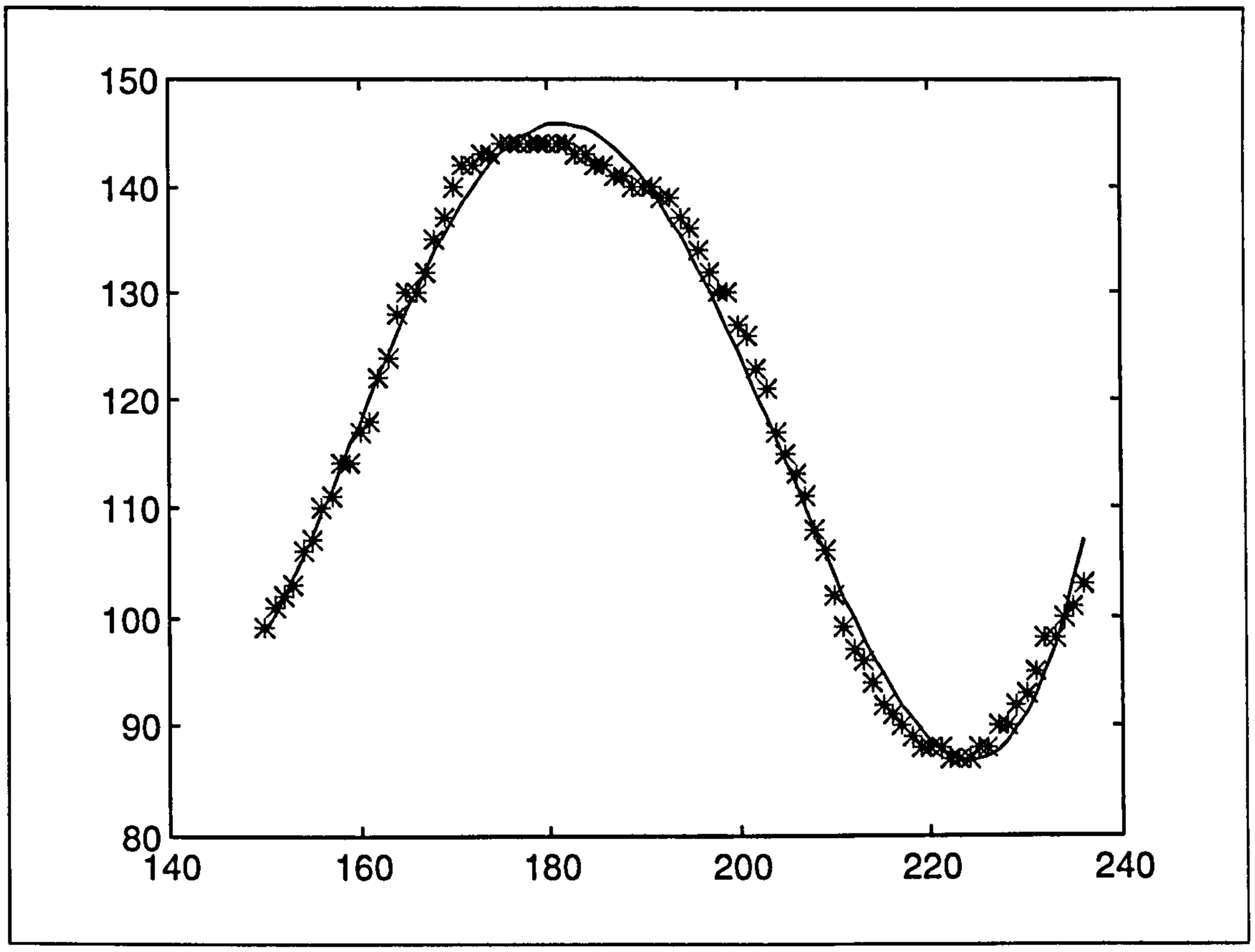


Figure 6.37e: Sixth order polynomial fit used to estimate fringe maximum point to sub-pixel accuracy

6.5.2 INTERPOLATION ROUTINE

This routine uses a least square fit to accurately detect the position of local fringe maxima present in an interferogram to sub-pixel accuracy. It uses a polynomial fit of the form:

$$I(X) = A_m X^m + A_{m-1}X^{m-1} + A_{m-2}X^{m-3} + A_{m-3}X^{m-4} + \ldots\ldots\ldots + A_0X^0 \qquad (6.32)$$

to interpolate the one-dimensional intensity profile of an interferogram, which is theoretically of the form (Gasvik, 1995 pp251):

$$I(X) = a + b\cos(\Delta\theta) = a + b\cos\left(\frac{2\pi X}{\lambda}\right) \qquad (6.33)$$

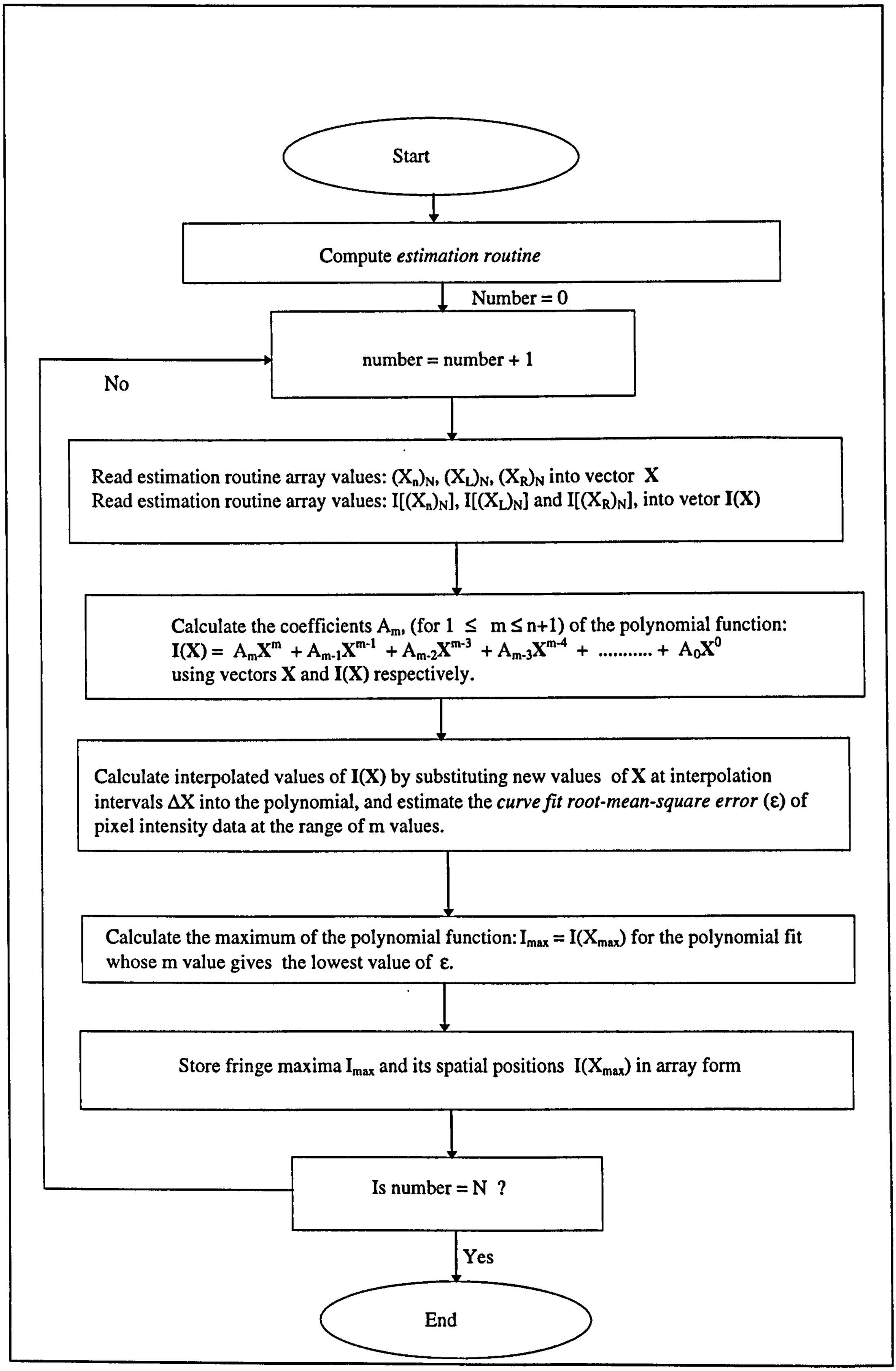


Figure 6.38: The *interpolation routine* flow diagram

where A_n: coefficients determined during the execution of the routine
m: order of polynomial
a, b: are the DC and AC terms of intensity profile
λ: wavelength of source
X^n: spatial positions of fringe intensity values
$\Delta\theta$: phase change
I(X): fringe intensity values (grey-levels)

The routine computes the values of A, B and C using the array values: $(X_n)_N$, $(X_a)_N$ and $(X_b)_N$ (obtained from the estimation routine), and their corresponding intensity values $I[(X_n)_N]$, $I[(X_a)_N]$ and $I[(X_b)_N]$ respectively.

Figure 6.37e shows an example of an interpolated intensity pattern (ie for the third fringe of the interferogram shown in figure 6.37a). It indicates the fringe maximum value (I_{max}) of 145.141 at the spatial position (X_{max}) of 181.340, these values were obtained by an iterative maximum point search routine:

$$\text{MAXIMUM}\{A_mX^m + A_{m-1}X^{m-1} + A_{m-2}X^{m-3} + \ldots\ldots\ldots + A_0X^0\} \qquad (6.34)$$

of the interpolated intensity function. A detailed description of the interpolation routine is given in figure 6.38.

Accurate measurement of fringe maxima in an interferogram depends on the interpolation interval (ΔX) and the order (m) of the polynomial fit. Tables 6.1 and 6.2 show *probable* maximum values and their estimated *root-mean-square error* (ε) for different values of m respectively for the third fringe in the interferogram shown in figure 3.37a. Table 6.3 suggests that the most accurate curve fit for the fringe data is a six order polynomial, as shown in figure 6.39. For most interferograms, polynomial fits of the $4 \leq m \leq n+1$ and $\Delta X = 0.1$ to 0.01 (ie up to a 100th of a pixel) will suffice, this equates to a fringe peak measurement accuracy of less than 6 nm for a nominal fringe spacing of 50 pixels (ie peak detection accuracy less than a pixel).

Original value $(X_3)_N$	Value at m = 1	Value at m = 2	Value at m = 3	Value at m = 4	Value at m = 5	Value at m = 6
1	2.000	23.420	20.380	20.380	21.600	21.960
69	70.000	103.250	97.460	101.090	101.070	99.510
150	151.000	182.840	176.680	181.410	181.340	181.970

Table 6.1: *Probable* maximum values at different values of m for the first 3 fringes

order of polynomial (m)	root-mean-square error
1	13.2053
2	6.9141
3	3.2732
4	0.8685
5	0.8353
6	0.8564

Table 6.2: Root-mean-square error of fringe data at different values of m

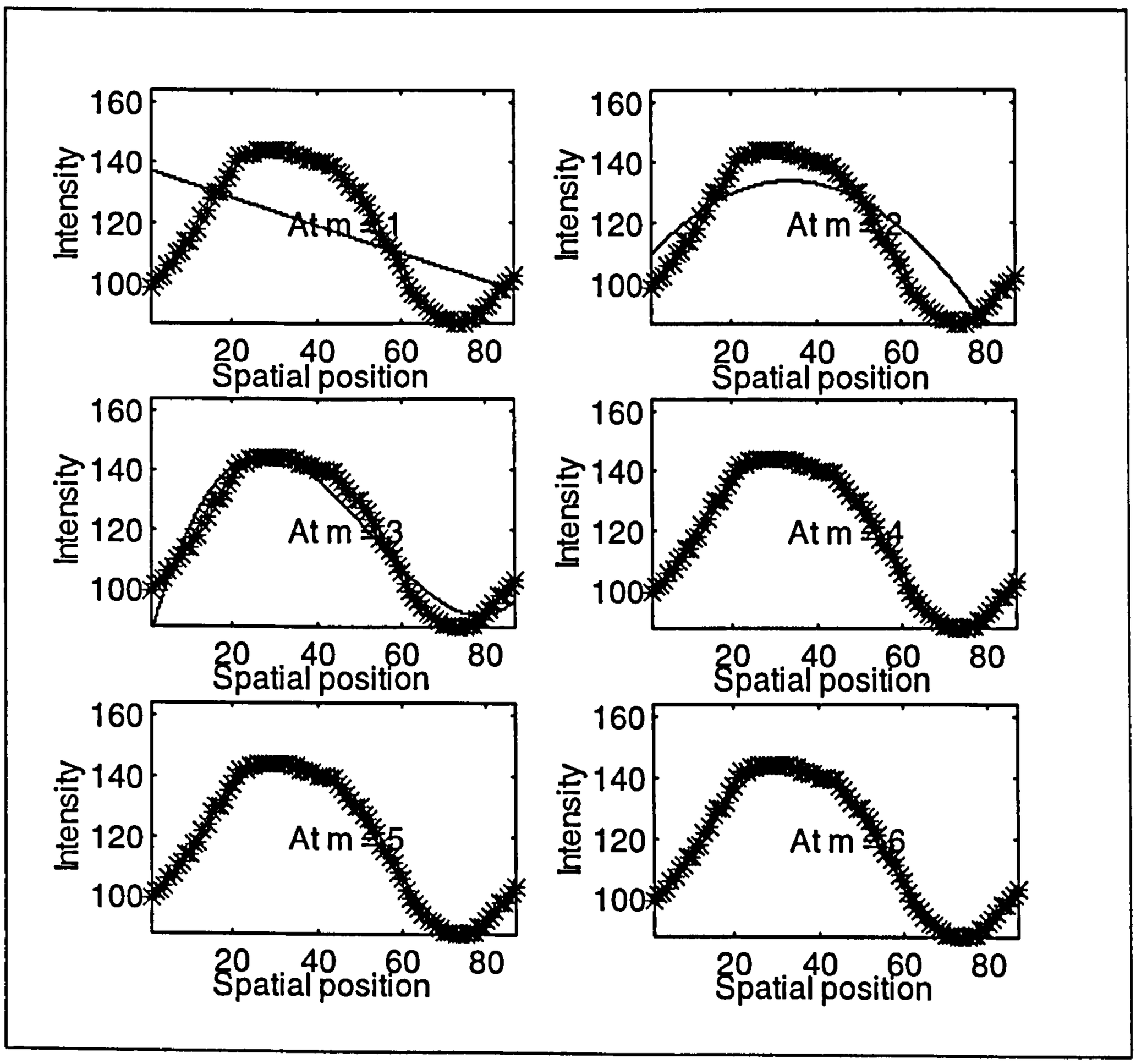


Figure 6.39: Polynomial fit to fringe data at different values of m

CHAPTER 7: INTERFEROMETER ADJUSTMENT

7.1 INTRODUCTION

As described in chapter 5, the spindle metrology system developed in this project incorporates an interferometer, a 2-axis servo-mechanism and image processing hardware. Automatic adjustment of the interferometer with the aid of the servo is required before the instrument can be used for spindle metrology.

Automatic adjustment of the interferometer is based on observed fringe dynamics and is partly satisfied when the interferometer *reference* surface normal is parallel with the spindle axis. Automatic adjustment of the interferometer is implemented by a number of control regimes: *on-off control* (with and without dither input), and *model reference adaptive control* (with dither).

In this chapter a detailed description of interferometer adjustment strategies and servo control are given. Also, a detailed analysis of servo non-linearity and techniques for reducing them are described. The software written for the servo control algorithms are given in the program listings booklet (accompanied with the thesis) in sections 2, 3 and 7.

7.2 INTERFEROMETER ADJUSTMENT CONTROL STRATEGIES

7.2.1 GEOMETRIC MODEL OF THE METROLOGY SYSTEM

Figure 7.1a shows the geometric model of the spindle metrology system in terms of axes and wavefront normals, it consists of spindle axis (**S**), workpiece surface normal (**W**) known as the workpiece axis, and interferometer reference surface normal (**I**) known as the interferometer axis. Figure 7.1b shows Cartesian co-ordinates X and Y lie in a plane normal to the median spindle axis and Z is coincident with it. $\mathbf{I}_{XY}$ is the projected component of **I** on the X-Y plane and forms ϕ_{XY} with the X axis, angles ϕ_I, ϕ_W and ϕ_O are between vectors **S** and **I**, **S** and **W** and **I** and **W** respectively.

In the ideal case, the spindle revolves about a single axis of rotation in the Z direction. However, in general the artefact **W** and interferometer **I** axes will not be aligned with each other or with the spindle. An offset tilt vector **P** is shown in figure 1b, representing the tilt between the axes of the interferometer **I** and artefact **W** i.e. governing the fringe pattern observed in the interferometer. The direction of this vector **P** corresponds to the orientation of the fringe pattern and its modulus corresponds to the fringe-pattern spatial frequency.

7.2.2 FRINGE DYNAMICS OBSERVED

In general, fringe spacing and orientation of a non-rotating workpiece surface is a function of the angle (ϕ_O) formed between the interferometer and workpiece axes,

CHAPTER 7: INTERFEROMETER ADJUSTMENT

7.1 INTRODUCTION

As described in chapter 5, the spindle metrology system developed in this project incorporates an interferometer, a 2-axis servo-mechanism and image processing hardware. Automatic adjustment of the interferometer with the aid of the servo is required before the instrument can be used for spindle metrology.

Automatic adjustment of the interferometer is based on observed fringe dynamics and is partly satisfied when the interferometer *reference* surface normal is parallel with the spindle axis. Automatic adjustment of the interferometer is implemented by a number of control regimes: *on-off control* (with and without dither input), and *model reference adaptive control* (with dither).

In this chapter a detailed description of interferometer adjustment strategies and servo control are given. Also, a detailed analysis of servo non-linearity and techniques for reducing them are described. The software written for the servo control algorithms are given in the program listings booklet (accompanied with the thesis) in sections 2, 3 and 7.

7.2 INTERFEROMETER ADJUSTMENT CONTROL STRATEGIES

7.2.1 GEOMETRIC MODEL OF THE METROLOGY SYSTEM

Figure 7.1a shows the geometric model of the spindle metrology system in terms of axes and wavefront normals, it consists of spindle axis (**S**), workpiece surface normal (**W**) known as the workpiece axis, and interferometer reference surface normal (**I**) known as the interferometer axis. Figure 7.1b shows Cartesian co-ordinates X and Y lie in a plane normal to the median spindle axis and Z is coincident with it. $\mathbf{I}_{XY}$ is the projected component of **I** on the X-Y plane and forms ϕ_{XY} with the X axis, angles ϕ_I, ϕ_W and ϕ_O are between vectors **S** and **I**, **S** and **W** and **I** and **W** respectively.

In the ideal case, the spindle revolves about a single axis of rotation in the Z direction. However, in general the artefact **W** and interferometer **I** axes will not be aligned with each other or with the spindle. An offset tilt vector **P** is shown in figure 1b, representing the tilt between the axes of the interferometer **I** and artefact **W** i.e. governing the fringe pattern observed in the interferometer. The direction of this vector **P** corresponds to the orientation of the fringe pattern and its modulus corresponds to the fringe-pattern spatial frequency.

7.2.2 FRINGE DYNAMICS OBSERVED

In general, fringe spacing and orientation of a non-rotating workpiece surface is a function of the angle (ϕ_O) formed between the interferometer and workpiece axes,

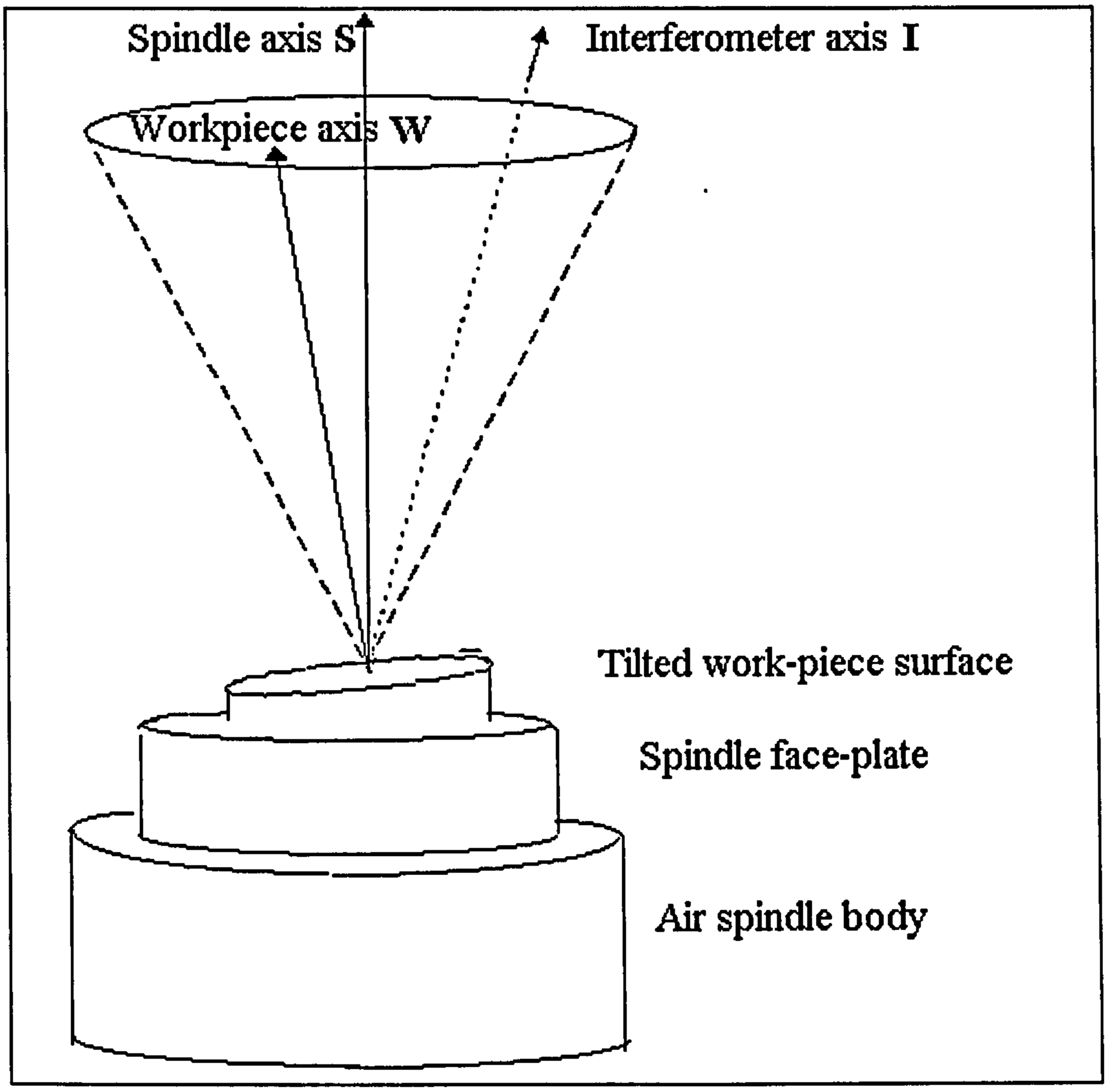


Figure 7.1a: Geometric model of metrology system

and by the pointing direction of the plane containing them, both of which are held constant for a measurement. Unlike this, in the case in hand, with the workpiece rotating about an axis not necessarily parallel to the interferometer axis, requires the inclusion of the spindle axis in order to describe fringe dynamics.

When the spindle and artefact rotate, the fringe pattern dynamics depends on the spindle angular position (θ), the values of ϕ_I, ϕ_W and ϕ_O, and the orientation of vectors **I**, **S** and **W** (Idowu & Gee, 1996). A number of fringe dynamic conditions analogous to static interferometry (Gee et al, 1988) may be visualised in terms of *general* cases:

- fringe pattern rotates but the fringe spacing varies cyclically ($\phi_W > \phi_I$)
- fringe pattern oscillates through partial rotations and the fringe spacing varies ($\phi_I > \phi_W$).

and *special* cases:

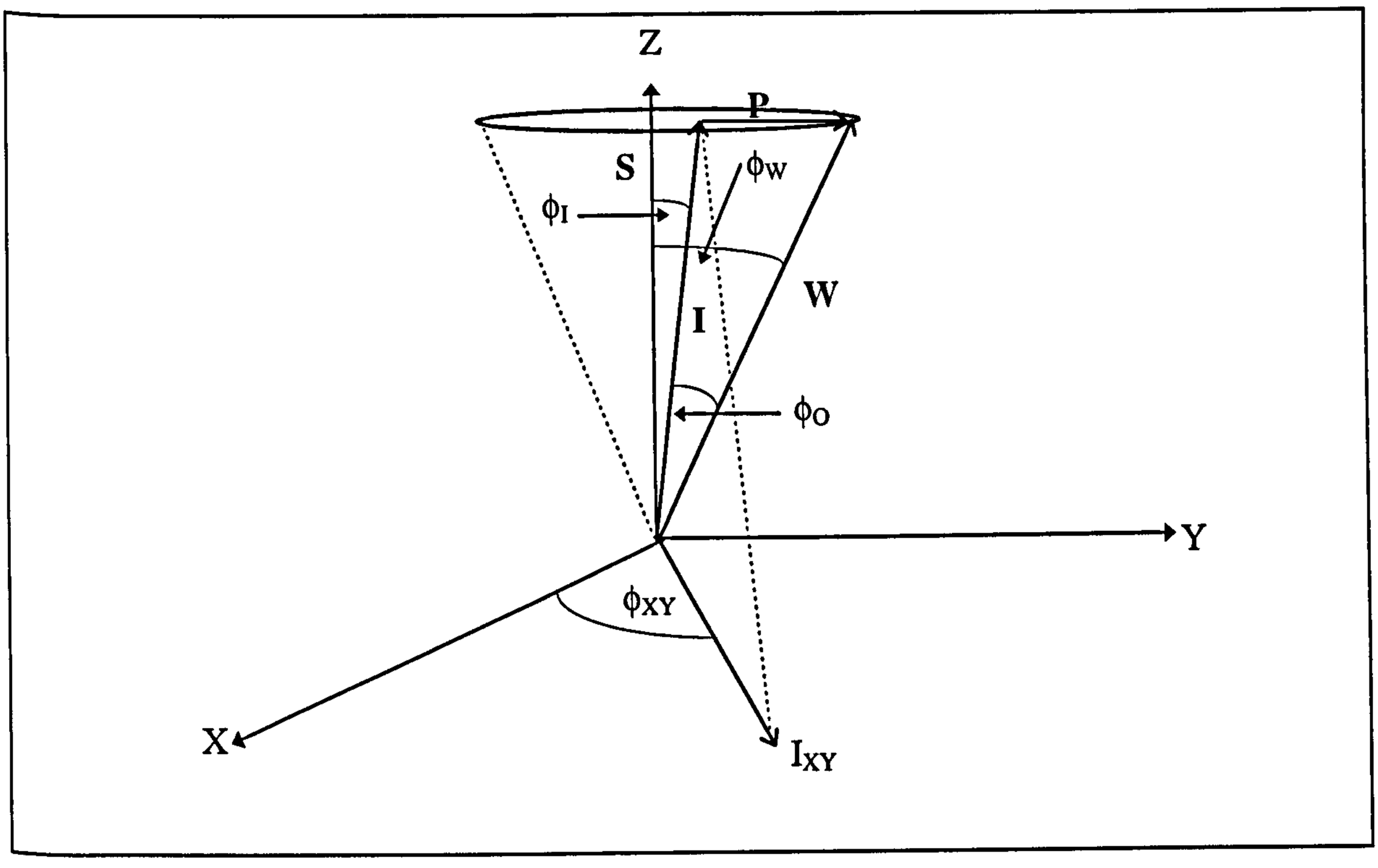


Figure 7.1b: Geometric model in details

- fringe pattern remain stationary as the spindle rotates ($\phi_W = 0$)
- fringes *fluff-out* to infinity at one point in spindle revolution ($\phi_W = \phi_I$)
- infinite fringe spacing ($\phi_W = \phi_I = 0$)
- fringes of constant spacing rotates at a constant rate ($\phi_I = 0$).

The orientations of the reference and workpiece axes can be described by the position of their point-spread-functions in the focal plane, with respect to the spindle axis of rotation (Gee et al, 1988). Figures 7.2a to 7.2f show plans view of the focal plane for different readily recognisable conditions, where the point spread functions of the focused interferometer beams correspond to the intersections of **I** and **W** with that plane denoted by *i* and *w* respectively. The spindle axis (**S**) intersects this plane at *s*. Vectors **P**, **Q** and **R** are respective tilts of the interferometer axis with respect to the workpiece axis, of the interferometer axis with respect to the spindle axis and of the workpiece axis with respect to the spindle axis.

7.2.3 INTERFEROMETER ADJUSTMENT

7.2.3.1 ANALYSIS OF INTERFEROMETER ADJUSTMENT

For spindle metrology, vector **P** is adjusted so that, for a perfect spindle, its magnitude remains constant as the artefact rotates. In this condition the interferometer reference surface normal **I** is parallel to the spindle axis **S**. This is done by adjusting the

interferometer axis into line with the spindle axis by reference to variations of the modulus and direction of vector **P** as the spindle revolves. Analytically, this occurs when the vector **P** is constant during rotation (ω) about the spindle axis ie when $\phi_I = 0$ and as $\phi_W = \phi_O$ (Idowu & Gee, 1996) shown in figure 7.2f.

In order to bring the interferometer axis into line with that of the spindle, a tilt-correction is required along the diameter defined by the positions of point-spread-functions $w_{\theta MAX}$ and $w_{\theta MIN}$ (shown schematically in figure 7.3a). Variations of the spatial frequency and orientation of vector **P** inherently cause angle ϕ_O to vary since:

$$\phi_O = \cos^{-1}\left(\frac{\mathbf{I}.\mathbf{W}}{I.W}\right) \qquad (7.1)$$

and $\mathbf{P} = \mathbf{W} - \mathbf{I}$ (7.2)

Two critical values of ϕ_O used for interferometer adjustment are:

$$\phi_{MAX} = \phi_I + \phi_W \qquad (7.3)$$
$$\phi_{MIN} = \phi_I - \phi_W \qquad (7.4)$$

The conditions in which these critical values are encountered can be identified by analysing *n* rotating fringe patterns at angular fixed intervals ($\Delta\theta$), using the algorithm described in section 6.5.

Figure 7.3a shows radial lengths P_{MIN} and P_{MAX} of the locus of *w*. These represent the maximum and minimum values of vector **P**. Figures 7.3b and 7.3c show the occurrence of maximum and minimum fringe densities during a typical interferometer adjustment. Maximum and minimum observed fringe densities P_{MIN} and P_{MAX} occur at spindle angles θ_{MAX} and θ_{MIN} respectively. ϕ_I is required to be zeroed in order to align the interferometer with the spindle. Adding equations (3) and (4), the required adjustment is:

$$\phi_I = \frac{(\phi_{MAX} + \phi_{MIN})}{2} = 0 \qquad (7.5)$$

and equation (7.5) is thus valid when: $\mathbf{P} = \mathbf{P}_{AV} = \frac{(P_{MAX} + P_{MIN})}{2}$ (7.6)

With a perfect spindle, when the interferometer is adjusted in accordance to equation (7.6), the fringe density P_{AV} remains constant and rotates at a constant rate with spindle revolution. A summary of the interferometer adjustment routine is shown in figure 7.4.

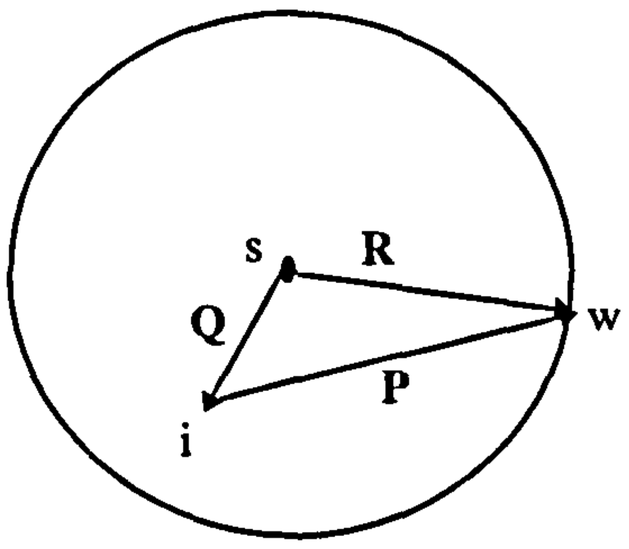


Figure 2a:General case $\phi_W = \phi_I$

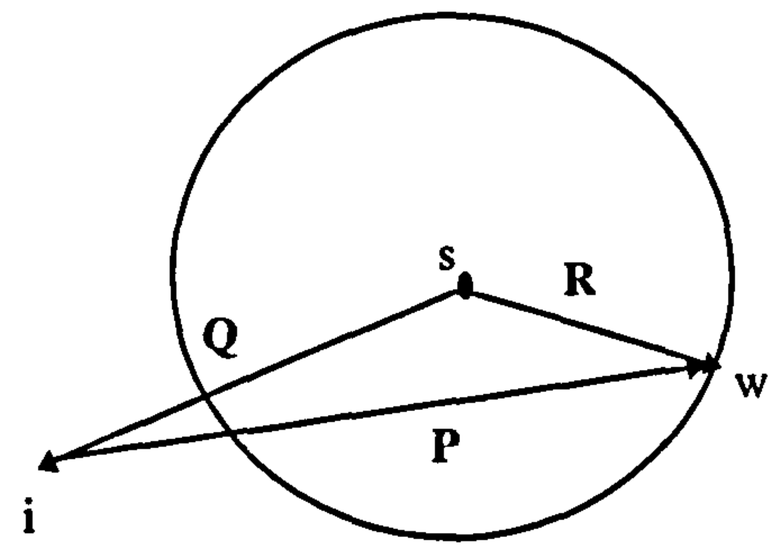


Figure 2b:General case $\phi_W < \phi_I$

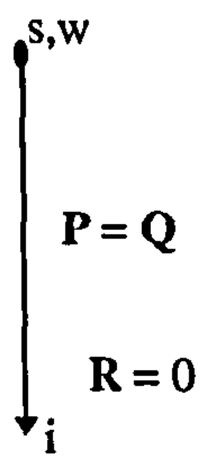


Figure 2c:Special case $\phi_I = 0$

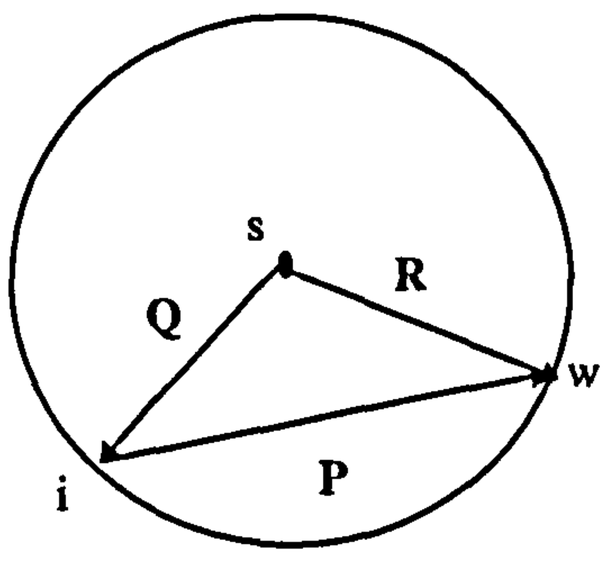


Figure 2d:General case $\phi_W = \phi_I$

s,i and w

P = R = Q = 0

Figure 2e:General case $\phi_W = \phi_I$

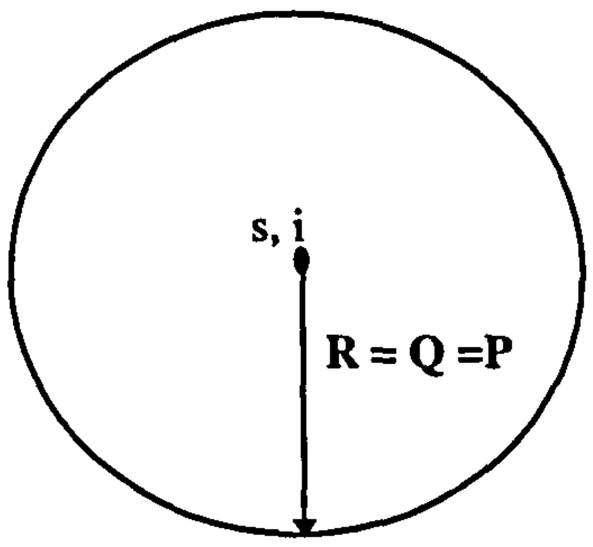


Figure 2f:General case $\phi_I = 0$

Figures 7.2a to 7.2f: Focal plane view of fringe dynamics

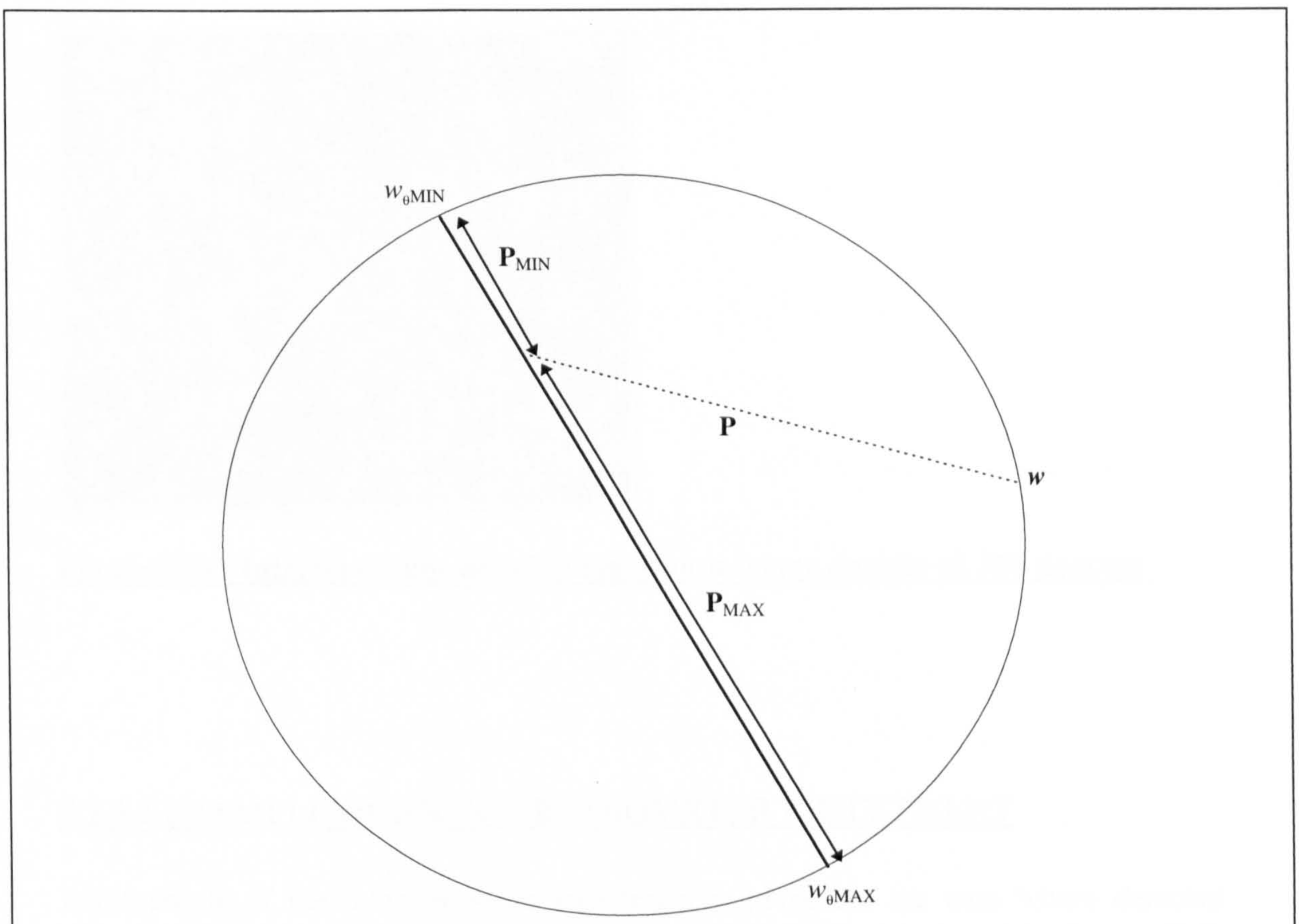


Figure 7.3a: Detection of maximum and minimum fringe density

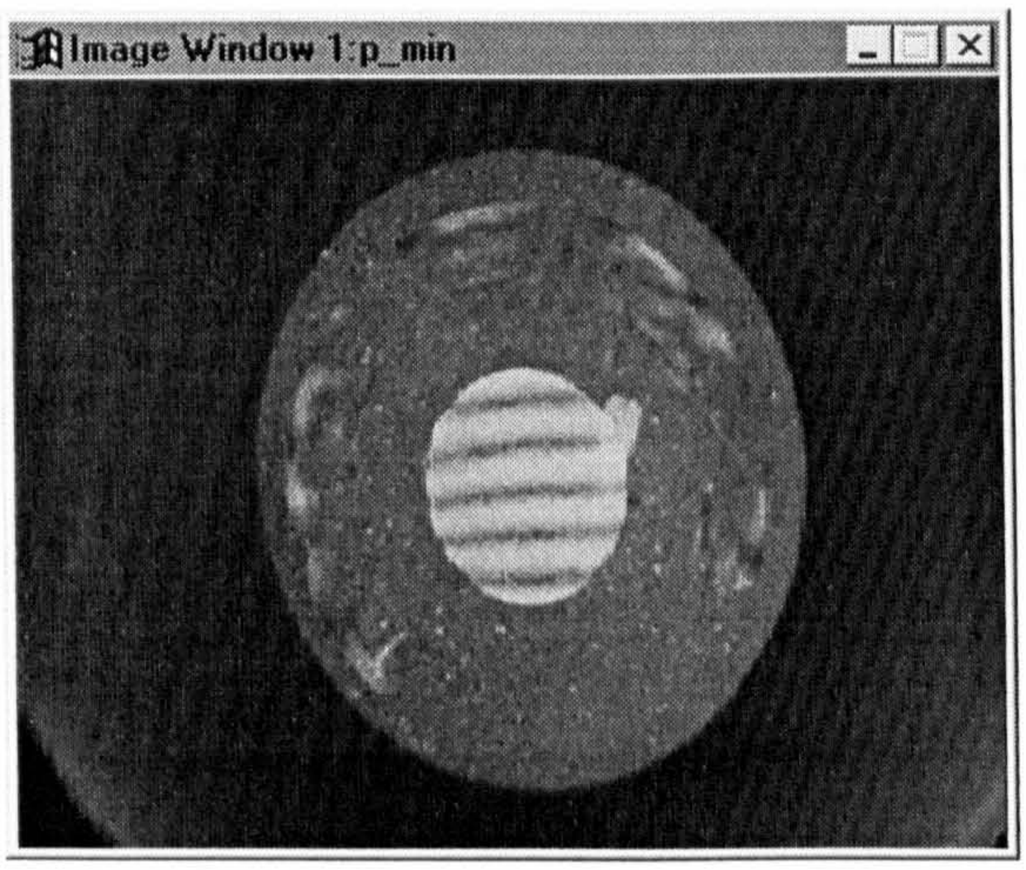


Figure 7.3b: Interferogram showing minimum fringe density at 180 degrees

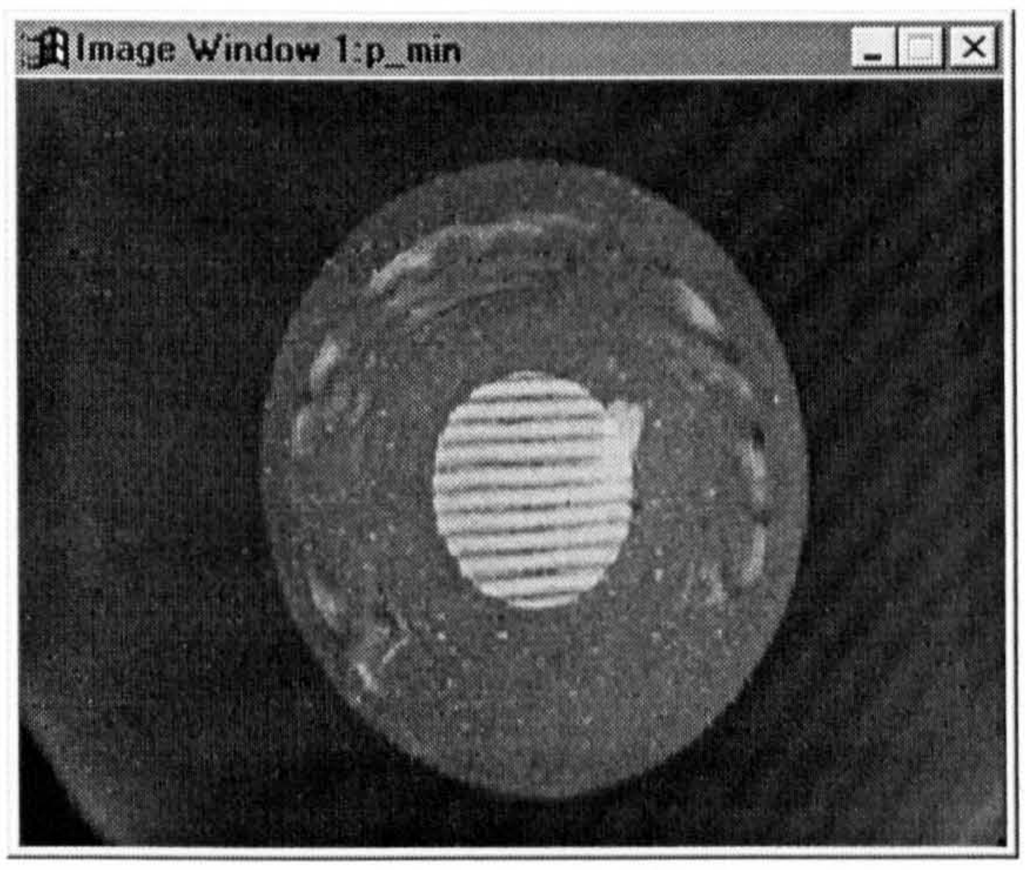


Figure 7.3c: Interferogram showing maximum fringe density at 360 degrees

7.2.3.2 EXAMPLE OF AN INTERFEROMETER ADJUSTMENT

An example of the interferometer adjustment is given for the case where detected values of P_{MIN} and P_{MAX} are 5 and 11 fringes respectively (as shown in figures 7.3b and 7.3c), and the values of θ_{MIN} and θ_{MAX} are 180° and 360° respectively. The required adjustment of the interferometer using its two-axis servo can be described mathematically with the aid of a Cartesian co-ordinate system shown in figure 7.5. The tilt vector **P** is proportional to the observed fringe density P and its orientation, therefore:

$$\mathbf{P_{MIN}} = P_{MIN}\left(\sin\theta_{MIN}\mathbf{i} + \cos\theta_{MIN}\mathbf{j}\right) \tag{7.7}$$

$$\mathbf{P_{MAX}} = P_{MAX}\left(\sin\theta_{MAX}\mathbf{i} + \cos\theta_{MAX}\mathbf{j}\right) \tag{7.8}$$

and, using equation (7.6), the required fringed density after interferometer adjustment is:

$$\mathbf{P}_{AV} = \frac{\left(P_{MAX}(\sin\theta_{MAX}\,\boldsymbol{i} + \cos\theta_{MAX}\,\mathrm{j}) + P_{MIN}(\sin\theta_{MIN}\,\boldsymbol{i} + \cos\theta_{MIN}\,\mathrm{j})\right)}{2} \tag{7.9}$$

The interferometer adjustment is achieved by applying a correction:

$$\mathbf{R} = M(\mathbf{P}_{MAX} - \mathbf{P}_{AV}) + C = M(\mathbf{P}_{MIN} + \mathbf{P}_{AV}) + C \tag{7.10}$$

where M and C are constants related to servo dynamics. Details of this are given in section 7.5.

For this particular example, assuming M = 1 and C = 0, the magnitude of the correction vector is:

$$\mathbf{R} = \left(11 - \frac{11+5}{2}\right)\sin 0\,\mathbf{i} + \cos 0\mathbf{j} = 3\,\mathbf{j}$$

This implies a tilt correction of 3 fringes is required, which is provided by the Y-axis servo in its positive direction.

7.2.3.3 SERVO SIGN CONVENTION

The servo used in this work is a two-axis mechanism which provides X and Y movements. Sign convention used for servo tilt correction vector (R) are:

X - axis servo	Y-axis servo	Sign of R
UP	RIGHT	+
DOWN	LEFT	-

Table 7.1: Servo sign convention

7.3 RELATIONSHIP BETWEEN FRINGE DENSITY, SPATIAL FREQUENCY AND SPINDLE ANGULAR POSITION

Interferometer adjustment requires knowledge of the fringe density (number of fringes present in an acquired interferogram frame) and spatial frequency during the rotation of the spindle. It is therefore important to understand the relationships between fringe density and spatial frequency, and fringe density and spindle angular position respectively.

The fringe spatial frequency (f) is essentially measured as the fringe density divided by the number of pixels per frame line (N). The nominal value of N in this work is 54 pixels. Hence f is directly proportional to P as shown graphically in figure 7.6 and

$$f = \frac{P}{N} \qquad (7.11)$$

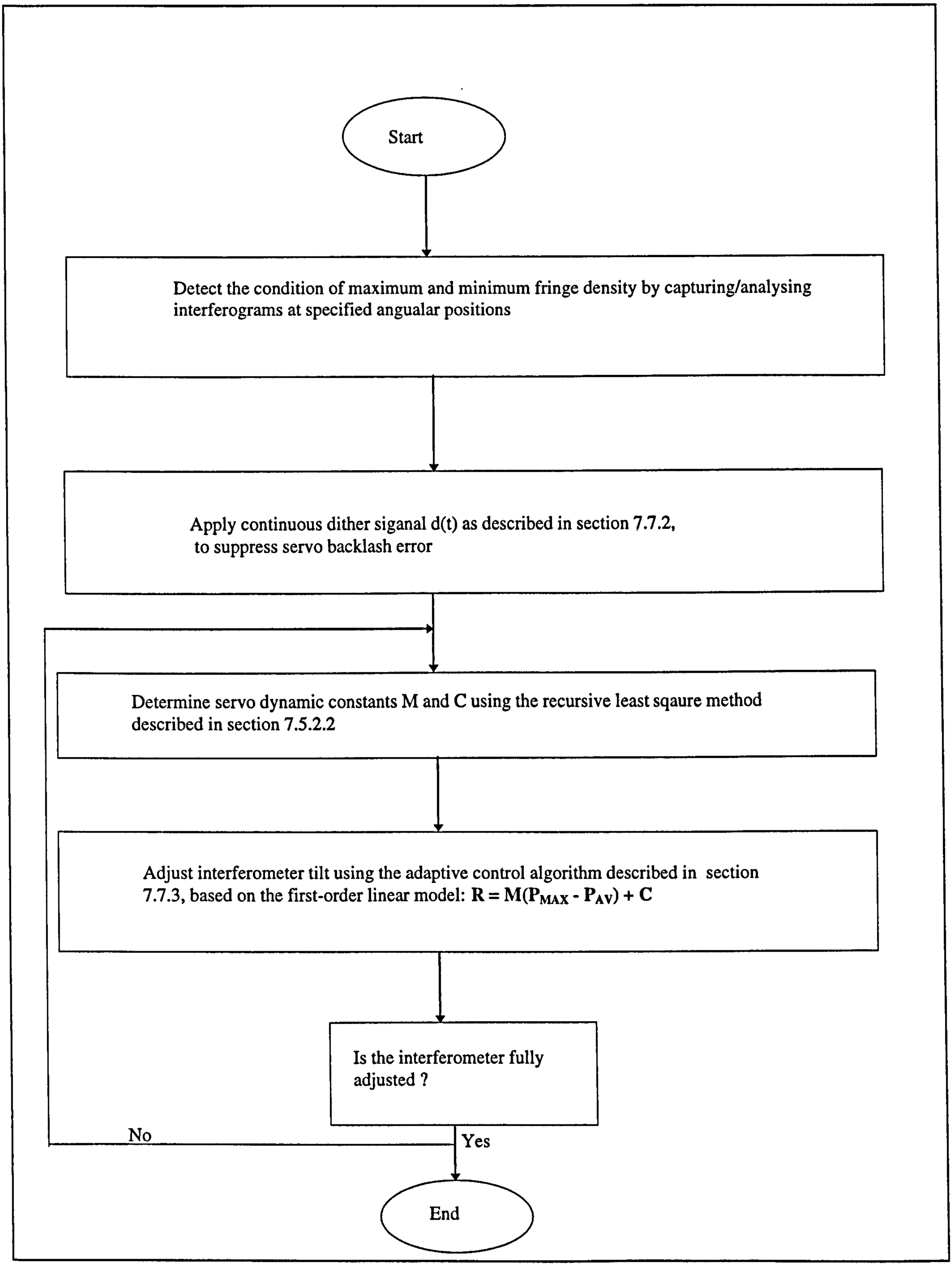


Figure 7.4: Interferometer adjustment routine

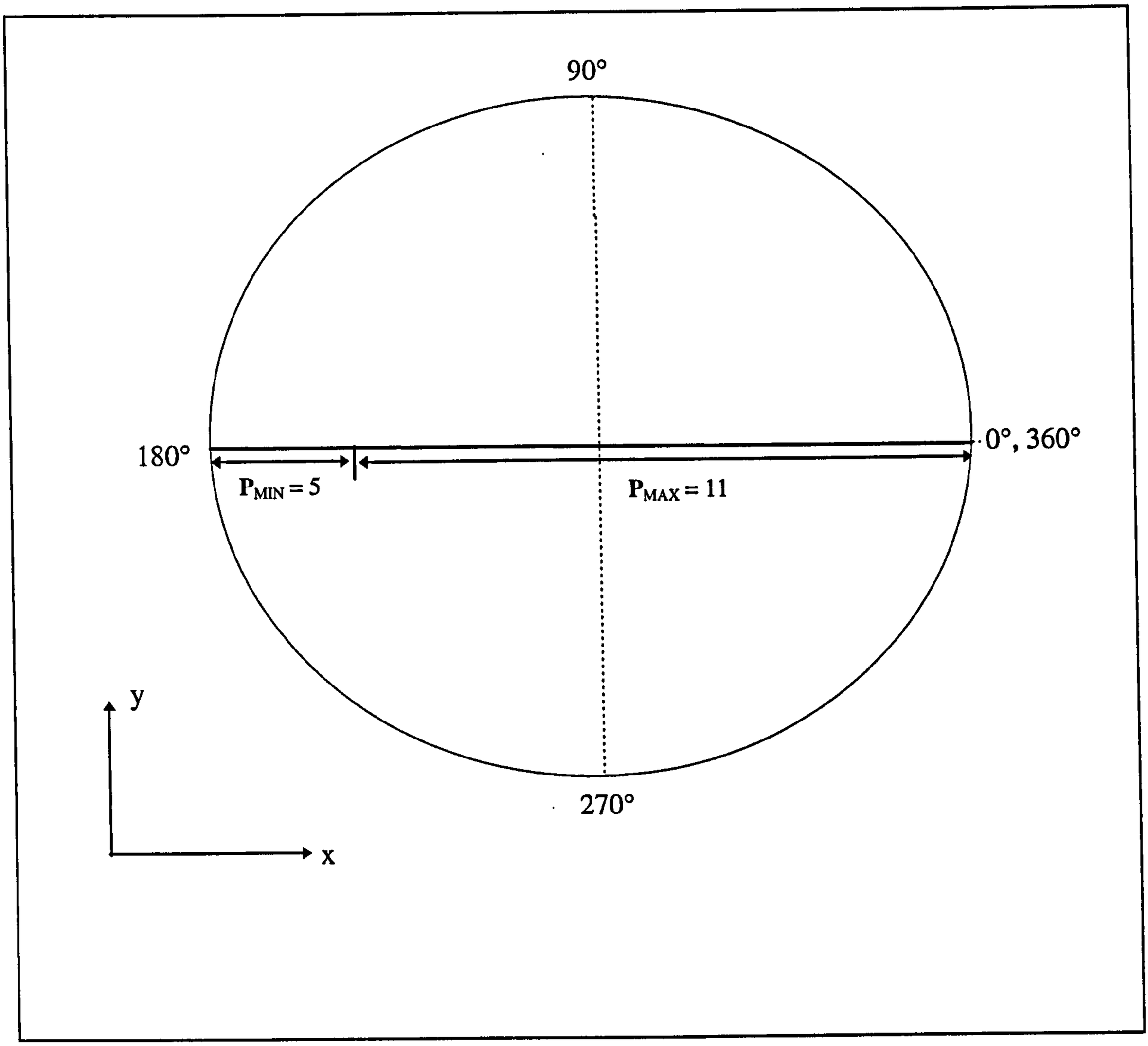


Figure 7.5: Cartesian system used to demonstrate interferometer control

The relationship between fringe density and spindle angular position is shown schematically in figure 7.7 and can be evaluated using the Cosine rule:

$$P = \{a^2 + b^2 - 2abcos(\beta + \theta)\}^{0.5} \qquad (7.12)$$

where, a: offset of interferometer point-spread-function from the spindle axis
b: offset of the work-piece point-spread-function from the spindle axis
β:angle between the vector **a** and the Y-axis
θ: spindle angular position

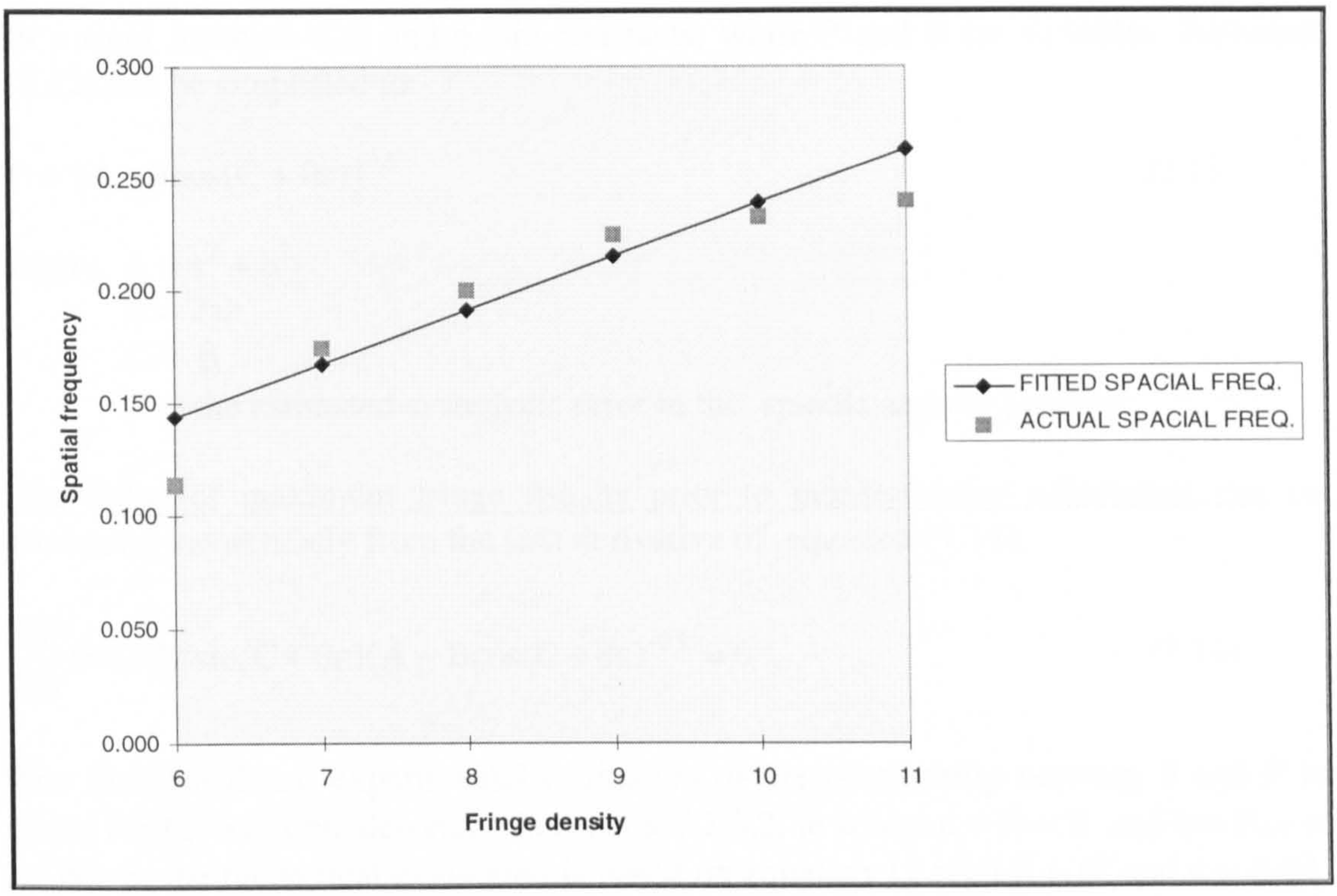


Figure 7.6: Linear relationship between fringe density (P) and spatial frequency (f)

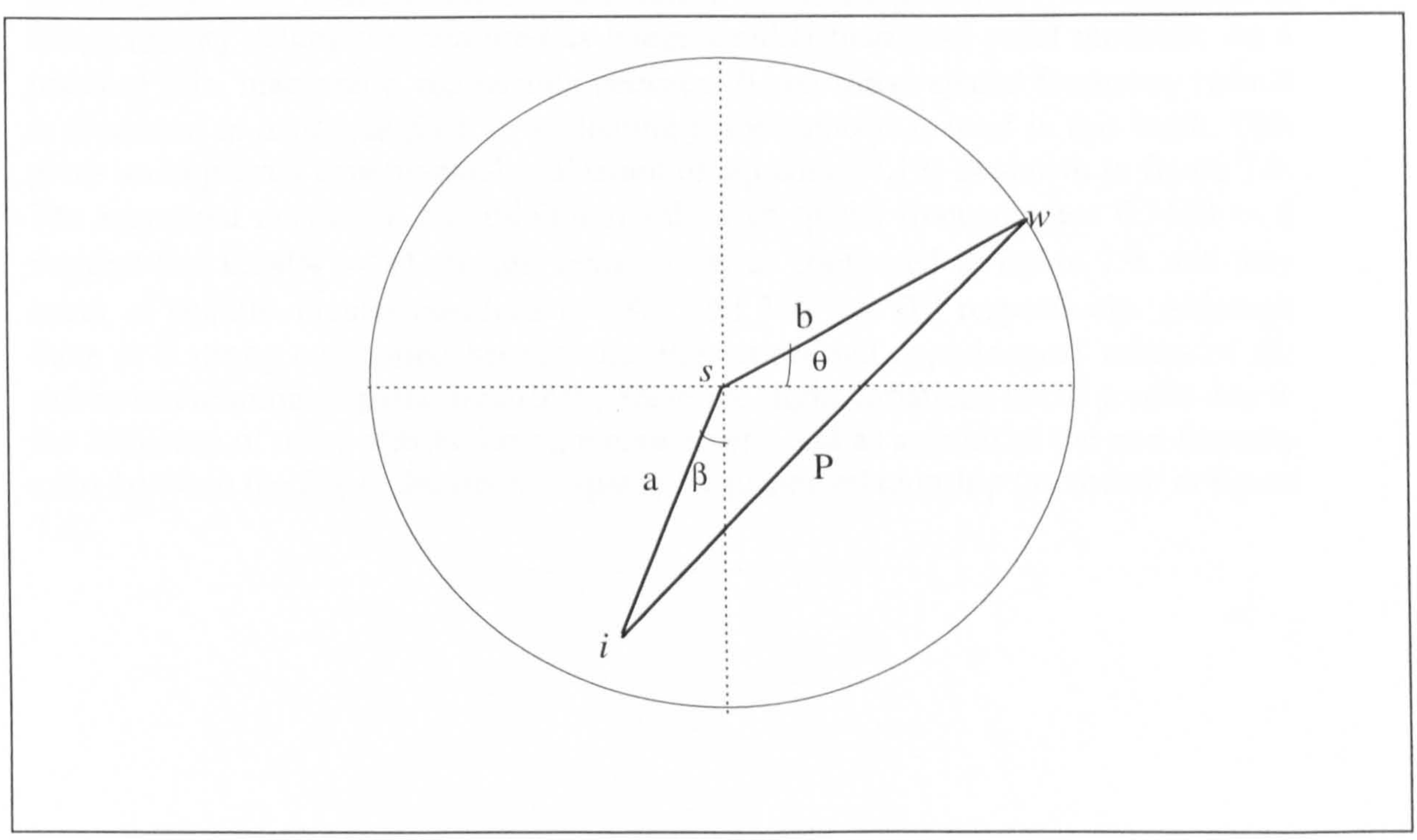


Figure 7.7: Schematic representation of the relationship between fringe density vector P and spindle angular position θ

In a rigid machine a, b and ϕ_I are constants, while P and θ are variables. Equation (7.12) can be simplified to:

$$P = \{A - B\cos(C + \theta\varepsilon)\}^{0.5} \qquad (7.13)$$

where, $A = a^2 + b^2$

$B = 2ab$

$C = \beta$

ε: is the estimated systematic error in the spindle angular position

The value of maximum fringe density prior to interferometer adjustment can be evaluated theoretically from the first derivative of equation (7.13):

$$\frac{dP}{d\theta} = 0.5B\varepsilon\sin(C+\theta\varepsilon)(A - B\cos(C+\theta\varepsilon)^{-0.5} = 0 \qquad (7.14)$$

The theoretical and experimental evaluation of the relationship between θ and P is given for the example described in section 7.2.3.2, ie where $a = R = 3$ and $b = P_{AV} = 8$. Where the theoretical evaluation is based on equation 12 with $\beta = 0°$ and $\varepsilon = 0.02$, while the experimental evaluation is obtained from the analysis of rotating interferograms.

Figure 7.8 shows the graphs of P against θ for both evaluations, indicating some discrepancy. This is due to the *discrete nature* of the fringe density measurement ie fringe density values are measured as integers rather than float point numbers. As a result of this inaccuracy, the relation between θ and fringe spatial frequency (which is measured in *analogue form* ie as floating point number) is used in this work. This gives an improved experimental evaluation of equation (7.13) as shown in figure 7.9. The measured maximum and minimum values of spatial frequency are 0.1133 ($\approx$ 5 fringes) and 0.2404 ($\approx$ 11 fringes) respectively as confirmed in figure 7.9. and they occur at spindle angular positions of 180° and 360° (or 0°) respectively. Although there is a strong correlation between the theoretical and experimental values of the maximum/minimum spatial frequency, there are slight variations in the graphs due to the influence of noise effects during measurement, and as a result of the non-linearity error between the fringe density and spatial frequency relationship (as shown in figure 7.6).

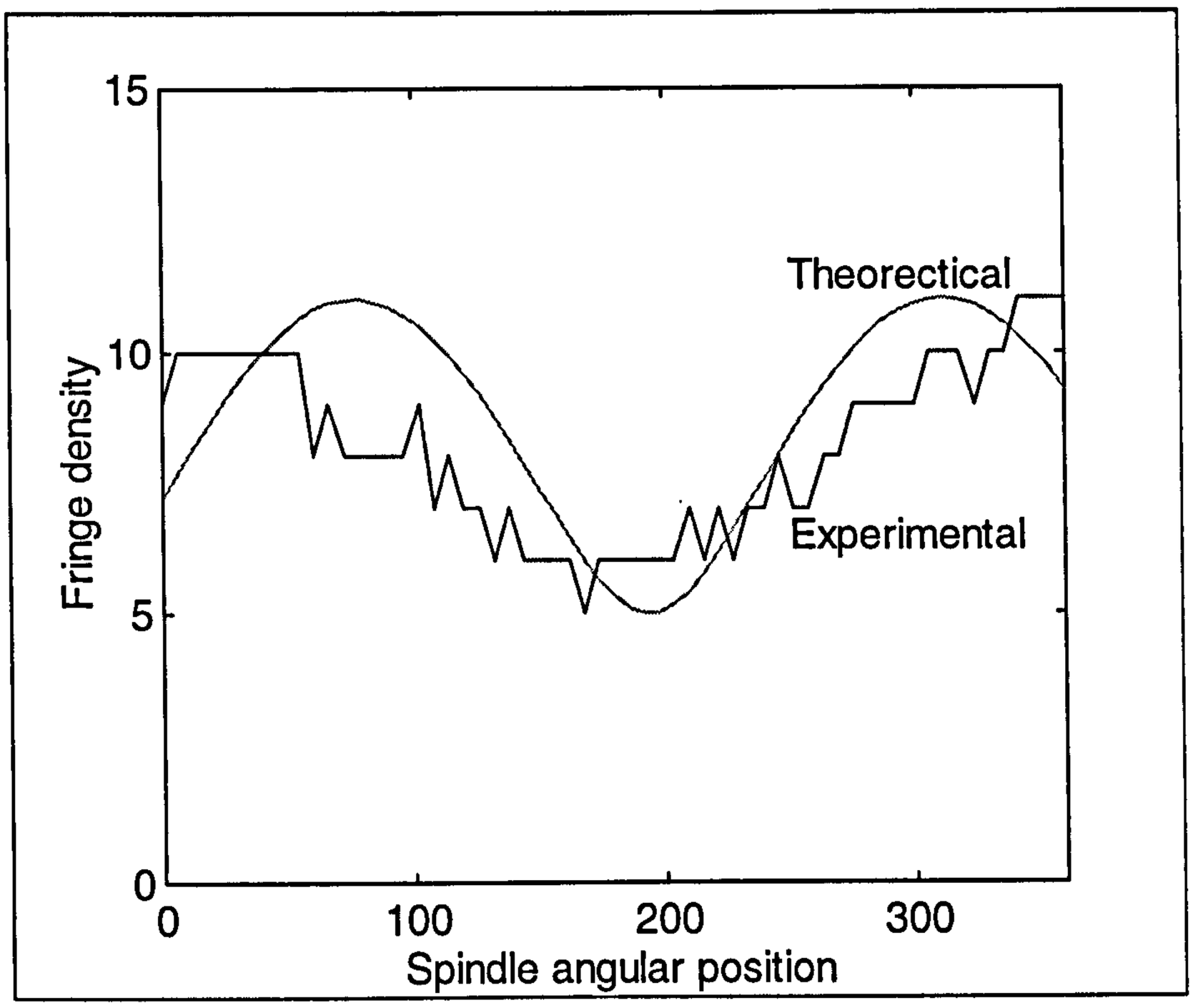


Figure 7.8: Experimental and theoretical relationship between P and θ

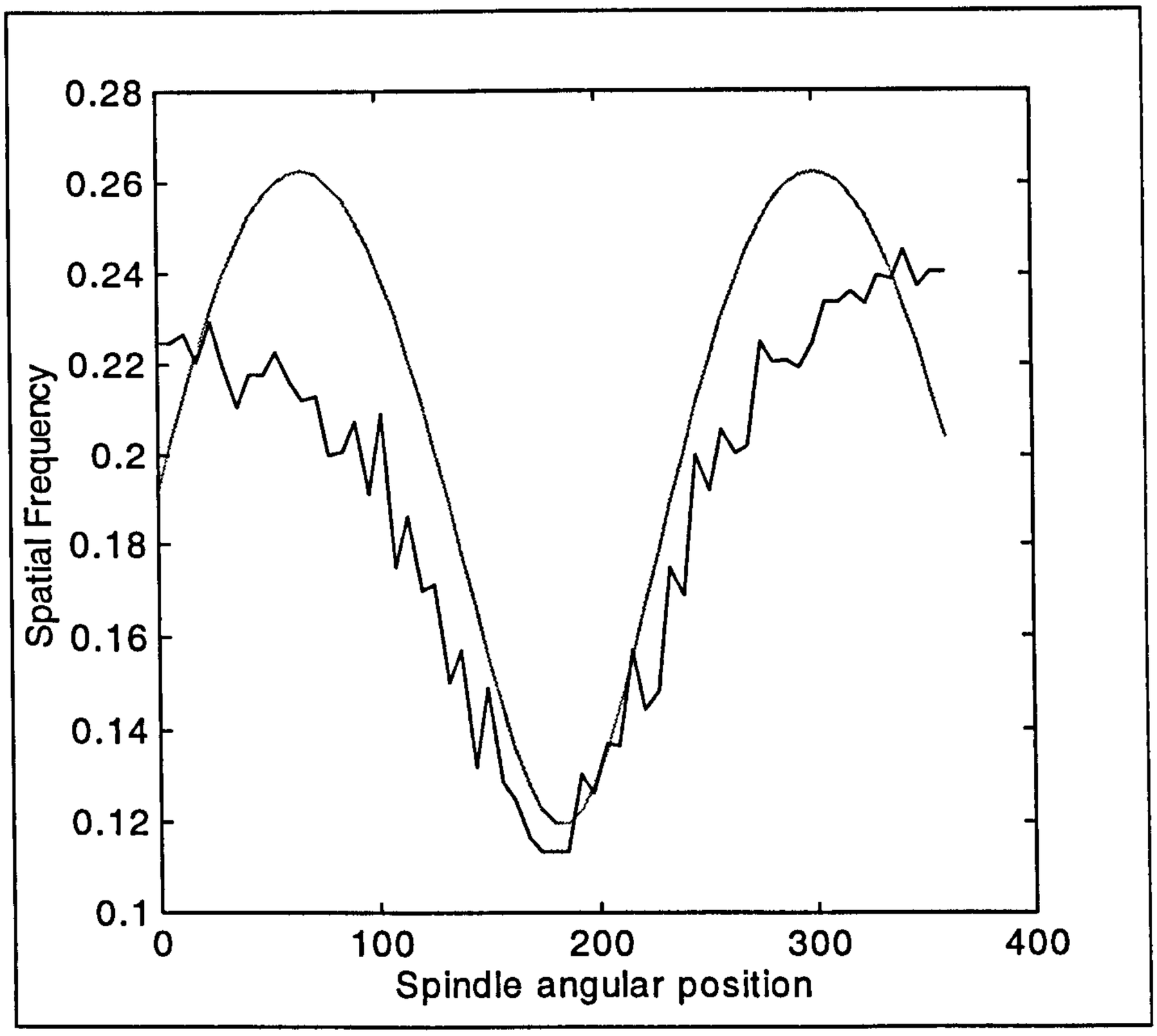


Figure 7.9: Experimental and theoretical relationship between f and θ

7.4 MATHEMATICAL MODEL OF INTERFEROMETER SERVO

7.4.1 INTRODUCTION

In this section a mathematical model of the two-axis interferometer servo-mechanism is given. The mechanism as shown schematically in figure 7.10 consists of the following components:

- electronic switching
- servo mechanics
- linear transmission
- interferogram processor
- unity feedback

In figure 7.10, input signals; $P_{sx}(s)$, $P_{sy}(s)$ are used to actuate the servo movements $X_x(s)$ and $Y_y(s)$ in order to adjust the interferometer by tilt rotation about its X and Y axes. Interferometer adjustment is confirmed from the measurement of fringe density values $P_x(s)$ and $P_y(s)$, which are used to reduce error signals $E_{rx}(s)$ and $E_{ry}(s)$ by feedback.

Construction details of the servo-mechanism, including a description of its components are given in section 5.5. In this work *first-order* and *second-order* mathematical models have been used to describe the servo characteristics.

7.4.2 EVALUATION OF SECOND-ORDER MODEL

The second-order model requires the transfer function of each servo component, this is used to find the resultant transfer function of the servo $G_s(s)$.

7.4.2.1 ELECTRONIC SWITCHING

An electronic switching unit is used by the servo to generate a series of low duration step pulses. A single step pulse can be defined mathematically as:

$$\begin{aligned} E(t) &= \chi E_r(t) \quad \text{for } t \le d \\ &= 0 \quad \text{for } t > d \end{aligned} \tag{7.14}$$

and the Laplace transform of $E_r(t)$ is: $E_r(s) = \chi/s$ (7.15)

where, $E_r(t)$: digital input signal [ie where $E_r(d) = 1$ and $E_r(0) = 0$]
t: time in seconds
d: duration of ON period of signal, nominally 1 second
χ: voltage gain value, nominally ± 12 V

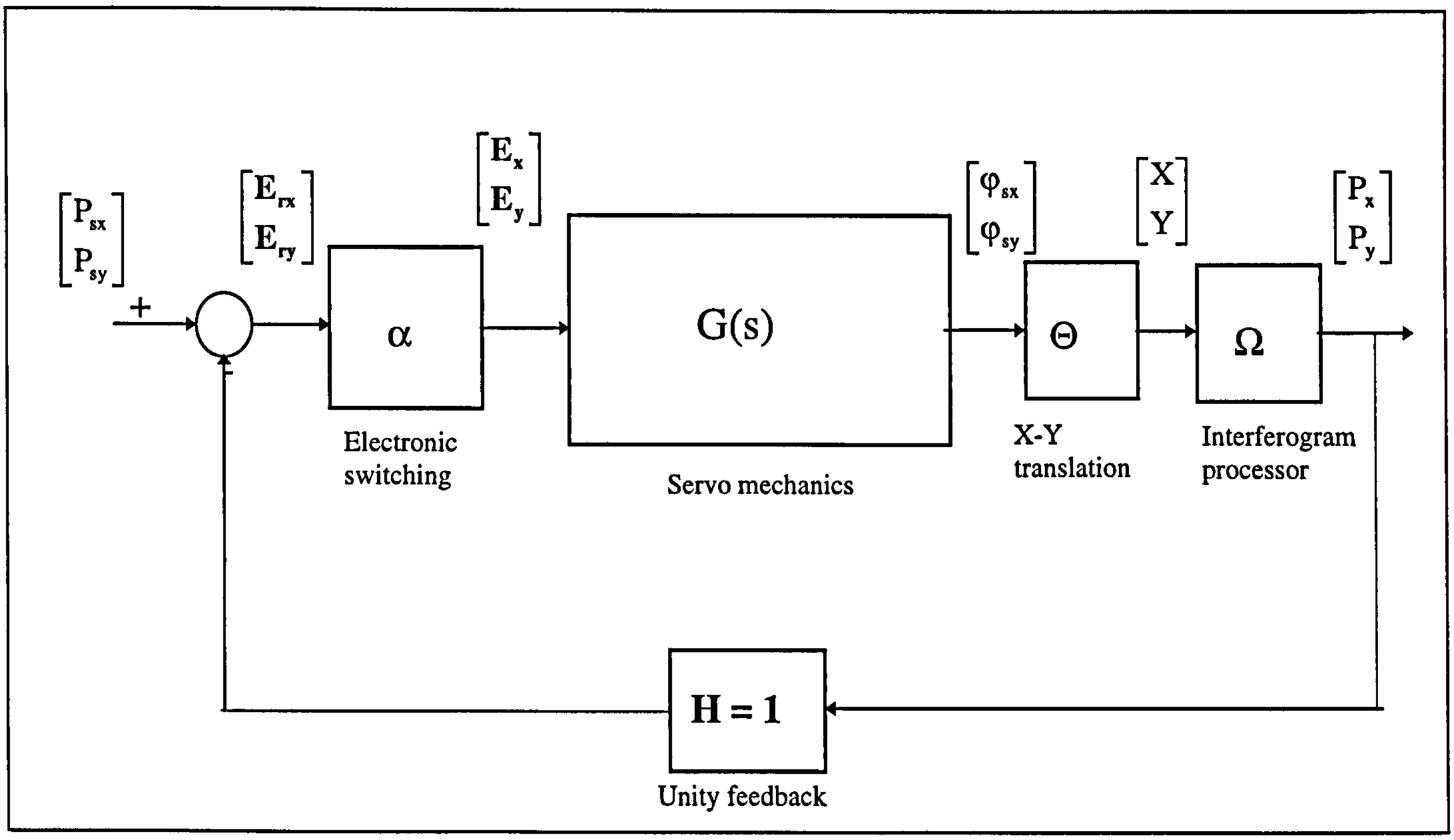


Figure 7.10: Servo mechanism and its components within a feedback loop

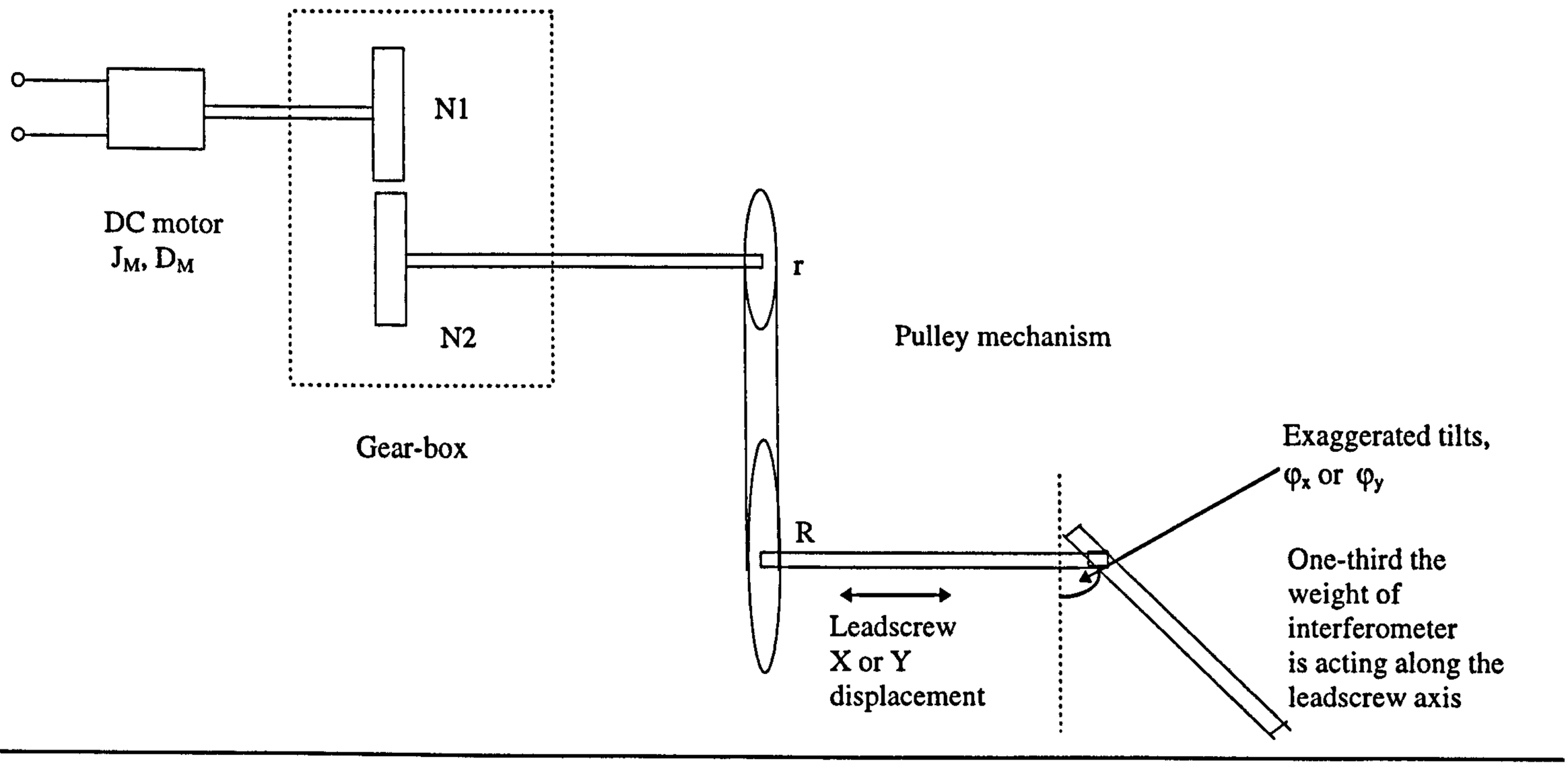


Figure 7.11: Schematic of Servo mechanism

7.4.2.2 SERVO MECHANISM

In this sub-section an evaluation of the transfer function of the servo is given. Both the X and Y servos used in this work consist of a DC motor, gear-box, pulley mechanism and lead-screw as shown schematically in figure 7.11.

The total inertia and damping values of the servo are:

$$\mathbf{J_T = J_M + J_L\left(\frac{N_1 r}{N_2 R}\right)^2} \qquad (7.16)$$

$$\mathbf{D_T = D_M + D_L\left(\frac{N_1 r}{N_2 R}\right)^2} \qquad (7.17)$$

respectively.

where, $\mathbf{J_M}$ and $\mathbf{J_L}$ are: inertia of servo motor and interferometer weight respectively
$\mathbf{D_M}$ and $\mathbf{D_L}$ are: damping values of servo motor and load respectively
N_1 and N_2 are: number of gear teeth of follower and driver respectively
R and r are: radius of the *follower* and *driver* pulley wheels respectively

From equations (7.16) and (7.17) the total torque generated by the servo (ie proportional to the first and second derivatives of the angular position (φ) its motor) is:

$$\mathbf{T_T(t) = J_T\ddot{\varphi}(t) + D_T\dot{\varphi}(t)} \qquad (7.18)$$

Lapalce transform of equation (18) is:

$$\mathbf{T_T(s) = (J_T s^2 + D_T s)\varphi(s)} \qquad (7.19)$$

The DC servo motor incorporates an armature resistance (R) and inductance (L), and is supplied with an input voltage **E(t)** which produces a back-emf (**V**) and current (**I**).

According to Sinha (1986, pp13-14), $\mathbf{E(t) = R\,I(t) + L\dot{I}(t) + V(t)}$ (7.20)
and the Laplace transform of equation (7.20) is:

$$\mathbf{E(s) = (Ls + R)\,I(s) + V(s)} \qquad (7.21)$$

The total torque and back-emf generated by the servo are:

$$\mathbf{T_T(s) = K_M\,I(s)} \qquad (7.22)$$

and $\mathbf{V(s) = sK_B\varphi(s)}$ (7.23)

Where K_M and K_B are the motor torque and back-emf constants respectively.

From equation (7.22), $\mathbf{I(s) = T_T(s) / K_M}$ (7.24)

Substituting equation (7.23) into (7.21) gives:

$$\mathbf{E(s)} = \frac{\mathbf{T_T(s)}}{\mathbf{K_M}}(\mathbf{R + Ls}) + \mathbf{V(s)} \qquad (7.25)$$

Substituting equation (7.19) and (7.23) into (7.25) gives:

$$\left(\frac{(\mathbf{J_T s^2 + D_T s})}{\mathbf{K_M}}(\mathbf{R + Ls}) + \mathbf{sK_B}\right)\varphi(\mathbf{s}) = \mathbf{E(s)} \qquad (7.26)$$

Equation (7.26) is the relationship between the applied input voltage **E(s)** to any one the servo arms, and their corresponding angular displacement $\varphi(\mathbf{s})$. In general the transfer function of the servo is:

$$\mathbf{G(s)} = \frac{\varphi(\mathbf{s})}{\mathbf{E(s)}} = \frac{\mathbf{b}}{\mathbf{s(s+a)}} \qquad (7.27)$$

where, $\mathbf{G(s)} = \begin{bmatrix} G_x(s) \\ G_y(s) \end{bmatrix}$: transfer function in vector form of X and Y servo respectively

$\varphi(\mathbf{s}) = \begin{bmatrix} \varphi_x(s) \\ \varphi_y(s) \end{bmatrix}$: angular displacement of X and Y servo arms respectively

$\mathbf{E(s)} = \begin{bmatrix} E_x(s) \\ E_y(s) \end{bmatrix}$: applied voltage to X and Y servo respectively

$\mathbf{D_T} = \begin{bmatrix} D_{Tx} \\ D_{Ty} \end{bmatrix}$: damping constant of X and Y servo respectively

$\mathbf{J_T} = \begin{bmatrix} J_{Tx} \\ J_{Ty} \end{bmatrix}$: inertial constant of X and Y servo respectively

$$\mathbf{a} = \frac{\left(\frac{\mathbf{K_M K_B}}{\mathbf{R+L}} + \mathbf{D_T}\right)}{\mathbf{J_T}} \qquad (7.28)$$

$$\mathbf{b} = \frac{\left(\frac{\mathbf{K_B}}{\mathbf{R+L}}\right)}{\mathbf{J_T}} \qquad (7.29)$$

a, b: are constants which relate to servo dynamics

7.4.2.3 X-Y TRANSLATION

The rotary motion generated by the servo lead-screws is converted to the required X-Y translation of the interferometer axis. X-Y translation of the interferometer is:

$$\begin{bmatrix} X(s) \\ Y(s) \end{bmatrix} = \begin{bmatrix} \varphi_x \\ \varphi_y \end{bmatrix} [\Theta] = \begin{bmatrix} \varphi_x \\ \varphi_y \end{bmatrix} \begin{bmatrix} \frac{L}{2\pi} \\ \frac{L}{2\pi} \end{bmatrix} \tag{7.30}$$

where, Θ: conversion vector of motion
L: pitch of the lead-screw

7.4.2.4 INTERFEROGRAM PROCESSOR

This component is used to quantify the density (P) of fringes orientated at β obtained during the X-Y translation of the interferometer, as the spindle rotates. It consists of a camera, frame-grabber, computer and interferogram analysis algorithm, details of this component are given in chapter 6.

Parameters P and β are proportional to the translation vector of the servo, such that:

$$\begin{bmatrix} P_x \\ P_y \end{bmatrix} = \begin{bmatrix} P(s)\sin\beta \\ P(s)\cos\beta \end{bmatrix} = \begin{bmatrix} X(s) \\ Y(s) \end{bmatrix} [\Omega] = \begin{bmatrix} X(s) \\ Y(s) \end{bmatrix} \begin{bmatrix} \frac{2}{\lambda} \\ \frac{2}{\lambda} \end{bmatrix} \tag{7.31}$$

where, $\Omega = \frac{\lambda}{2}$

λ: wavelength of interferometer light source
P_x, P_y: fringe density X and Y components

7.4.2.5 FULL MODEL OF SERVO

A combination of the servo-mechanism components described in sections 7.4.2.1 - 7.4.2.4, gives an overall second-order transfer function of:

$$G_s(s) = \frac{P(s)}{E_r(s)} = \chi G(s) \Theta \Omega \tag{7.32}$$

$$G_s(s)=\begin{bmatrix}G_x(s)\\G_y(s)\end{bmatrix}=\begin{bmatrix}\frac{P_x(s)}{E_{rx}(s)}\\\frac{P_y}{E_{ry}(s)}\end{bmatrix}=\begin{bmatrix}\frac{\chi\left(\frac{L}{2\pi}\right)\left(\frac{2}{\lambda}\right)b_x}{s(s+a_x)}\\\frac{\chi\left(\frac{L}{2\pi}\right)\left(\frac{2}{\lambda}\right)b_y}{s(s+a_y)}\end{bmatrix}=\begin{bmatrix}\frac{B_x}{s(s+A_x)}\\\frac{B_y}{s(s+A_y)}\end{bmatrix} \quad (7.33)$$

Dynamic behaviour of the servo along its X and Y axes depends on the values of constants A_x and B_x, and A_y and B_y respectively. Equations (7.28) and (7.29) show that these constants depend on the inertial and damping characteristics of the servo. In order to adjust the interferometer with the servo a knowledge of theses constants is required.

7.4.3 EVALUATION OF FIRST-ORDER MODEL

The first-order model of the interferometer servo is a simplified representation (approximation) of the second-order model given in equation 32 , it has the form:

$$G_s(s)=\frac{P(s)}{E_r(s)}=\chi s(s)\Theta\Omega+\frac{C}{E_r(s)}=\frac{M}{s}+\frac{C}{E_r(s)} \quad (7.34)$$

Where, $\mathbf{M}=\begin{bmatrix}M_x\\M_y\end{bmatrix}$: slope of linear function

$\mathbf{C}=\begin{bmatrix}C_x\\C_y\end{bmatrix}$: Y-intercept of linear function

Equation (7.34) represents a linear (ie straight-line fit) relationship between the fringe density (**P**) and applied input pulse ($\mathbf{E_r}$), where slope (**M**) and Y-intercept (**C**) are the required servo dynamic constants.

7.5 DETERMINING SERVO CONSTANTS

7.5.1 INTRODUCTION

There are a number of techniques used for identifying the dynamic behaviour of a servos mechanism, these include:

- Off-line methods: such as frequency response, step response, pseudo random binary signal response and generalised least-square (see for example, Sinha & Kuszta 1983, Phillips & Nagle 1990)
- In-situ methods: such as recursive least-square (see for example, Phillips & Nagle 1990, Iserman et al 1992, Astrom & Witternmark 1995) and correlation.

The process of identifying dynamic constants of a system is generally referred to as *system identification*. In this project system identification has been used:

- to identify (time-invariant) constants A_x, B_x, A_y and B_y of the second-order model using the *generalised* least- square method
- to identify (time-invariant) constants M_x, K_x, M_y and K_y of the first-order model using the *generalised* least-square method.
- to identify (time-varying) constants M_x, K_x, M_y and K_y of the first-order model using the *recursive* least-square method.

7.5.2 IDENTIFYING CONSTANTS - GENERALISED LEAST-SQUARE

The generalised least-square method uses the servo input voltage $\mathbf{E_r(s)}$ and the detected fringe density data **P(s)** to evaluate servo dynamic constants. This method is essentially an *off-line* method, meaning that servo dynamic constants are not evaluated in real-time. However, this will suffice for a servo which is linear and time-invariant, (ie see section 7.6 for details on servo linearity and its variation in time).

Application of this method requires:

- order (n) of the servo to be known a piori; for the first and second-order models n is 1 and 2 respectively
- equi-spaced sample of input and output data (ie shown in table 7.2) are required

The generalised least-square approach involves the following sequence of steps (Sinha & Kuszta 1983):

- Obtain experimental data samples (t) of servo in the form of step input data $e_r(t)$ and its output response p(t)
- Convert servo transfer function equation from s (continuous time-domain) to z (discrete-time domain)
- Evaluate servo dynamic constants in the z-domain
- Evaluate servo dynamic constants in the s-domain using the values obtained in latter step

7.5.2.1 IDENTIFYING CONSTANTS - SECOND-ORDER MODEL

The second-order transfer function of the servo given in equation (7.33) in its generalised form is:

$$\mathbf{G_s(s) = \frac{P(s)}{E_r(s)} = \frac{B}{s(s+A)} = \frac{B}{A}\left(\frac{1}{s} - \frac{1}{s+A}\right)} \qquad (7.35)$$

Equation (7.35) in its discrete form ie its conversion from s to z domain [see for example undergraduate texts on discrete control such as Sinha (1986, pp 259 -279)] :

$$\mathbf{G_s(z) = \frac{B}{A}\left(\frac{z}{z-1} - \frac{z}{z-e^{-AT}}\right) = \left(\frac{z^{-1}a_3}{1-z^{-1}a_1 - z^{-2}a_2}\right)} \qquad (7.36)$$

where, $z = e^{st}$: discrete domain representation

$\mathbf{a_1} = (1+e^{-At})$: dynamic constant in z-domain (7.37)

$\mathbf{a_2} = -e^{-At}$: dynamic constant in z-domain (7.38)

$\mathbf{a_3} = \frac{B}{A}\left(1 - e^{-At}\right)$: dynamic constant in z-domain (7.39)

t: sampling interval in seconds

Equation (7.36) in difference equation form is:

$$\mathbf{p(t) = a_1 p(t-1) + a_2 p(t-2) + a_3 e_r(t-1)} \qquad (7.40)$$

Time interval t (seconds) t = k	Step input voltage $e_r(k)$ (volts)	Fringe density output X-axis $p_x(k)$	Fringe density output Y-axis $p_y(k)$
0	0	10	9
1	1	10	8
2	1	9	7
3	1	8	6
4	1	7	6
5	1	6	5
6	1	6	5
7	1	5	6
8	1	5	4
9	1	3	2

Table 7.2: Servo response to step inputs

Equation (7.40) can expressed as:

$$\mathbf{p(N) = f(N)\mu + h(N)} \qquad (7.41)$$

where, N: number of data samples

p : fringe density data set

f : input and output data set matrix

$\mu = [a_1\ a_2\ a_3, \ldots\ldots a_{n+1}]$: unknown dynamic constants in z-domain

h : noise present in the servo output data

Using the generalised least-square method (Phillips & Nagle, 1990) the values (μ_e) of the servo dynamic constants is:

$$\mu_e = [\mathbf{f}^T(N)\mathbf{f}(N)]^{-1}\,\mathbf{f}^T(N)\mathbf{p}(N) \tag{7.42}$$

Using the data given in table 7.2 (ie this data is the input/output data of the X and Y axis servos), the variables **f**(N) and **p**(N) are:

$$\mathbf{f} = \begin{bmatrix} 10 & 10 & 0 \\ 10 & 9 & 1 \\ 9 & 8 & 1 \\ 8 & 7 & 1 \\ 7 & 6 & 1 \\ 6 & 6 & 1 \\ 6 & 5 & 1 \\ 5 & 5 & 1 \end{bmatrix} \qquad \mathbf{p} = \begin{bmatrix} 9 \\ 8 \\ 7 \\ 6 \\ 6 \\ 5 \\ 5 \\ 3 \end{bmatrix} \tag{7.43}$$

The dynamic constant of the X-axis servo in the z domain based on equation (42) is:

$$\begin{bmatrix} a_{x1} \\ a_{x2} \\ a_{x3} \end{bmatrix} = \begin{bmatrix} -1.0697 \\ -0.1828 \\ 0.8778 \end{bmatrix} \tag{7.44}$$

From equations (7.37) to (7.39), the dynamic constants in the s-domain are:

$$\begin{bmatrix} A_x \\ B_x \end{bmatrix} = \begin{bmatrix} 1.6993 \\ 0.3337 \end{bmatrix} \tag{7.45}$$

Hence, the transfer function of the X-axis servo is:

$$\mathbf{G_x}(s) = \frac{0.3337}{s(s+1.6993)} \tag{7.46}$$

Similarly, the transfer function of the Y-axis servo is:

$$\mathbf{G_y}(s) = \frac{0.012}{s(s+2.576)} \tag{7.47}$$

7.5.2.2 IDENTIFYING CONSTANTS - FIRST-ORDER (APPROXIMATION) MODEL

The first-order transfer function of the servo given in equation (7.34) in its generalised form is:

$$\mathbf{G_s(s) = \frac{P(s)}{E_r(s)} = \frac{M}{s} + \frac{C}{E_r(s)}} \tag{7.46}$$

$$\mathbf{p(t) = M \sum_{t=0}^{t=N} e_r(t) + C} \tag{7.47}$$

where, **p**: fringe density data set

$\mathbf{e_r}$: input pulse data set

$\mathbf{M} = \begin{bmatrix} M_x \\ M_y \end{bmatrix}$: slope of model linear function

$\mathbf{C} = \begin{bmatrix} C_x \\ C_y \end{bmatrix}$: Y-intercept of model linear function

t : pulse on time

N: pulse time at last data set

Using data given in table 7.2 and equation (7.42) the dynamic constants of the X-axis servo are:

$$\begin{bmatrix} M_x \\ C_x \end{bmatrix} = \begin{bmatrix} -0.7576 \\ 11.0667 \end{bmatrix}$$

Similarly the dynamic constants of the Y-axis servo are:

$$\begin{bmatrix} M_y \\ C_y \end{bmatrix} = \begin{bmatrix} -0.6061 \\ 9.3334 \end{bmatrix}$$

The generalised least-square method has been improved to cope identifying servo dynamic constants **M** and **C** in real-time, using a recursive least square method. This method has been used to characterise the time-varying nature of servo by recursively updating the linear relationship between the fringe density and input pulse. The flow charts shown in figures 7.12a and 7.12b describe the least square algorithm.

7.6 DETERMINATION OF SERVO LINEARITY

The relationship between the unit step input voltage $\mathbf{e_r}(t)$ and fringe density $\mathbf{p}(t)$ is ideally linear. However, this characteristic is usually perturbed by non-linear effects such as lead-screw backlash ($\mathbf{e_b}$) and stochastic noise ($\mathbf{e_n}$) in the fringe density measurement due to random fluctuations of temperature and pressure (which cause fringe drift), and occasional stiction of leadscrew.

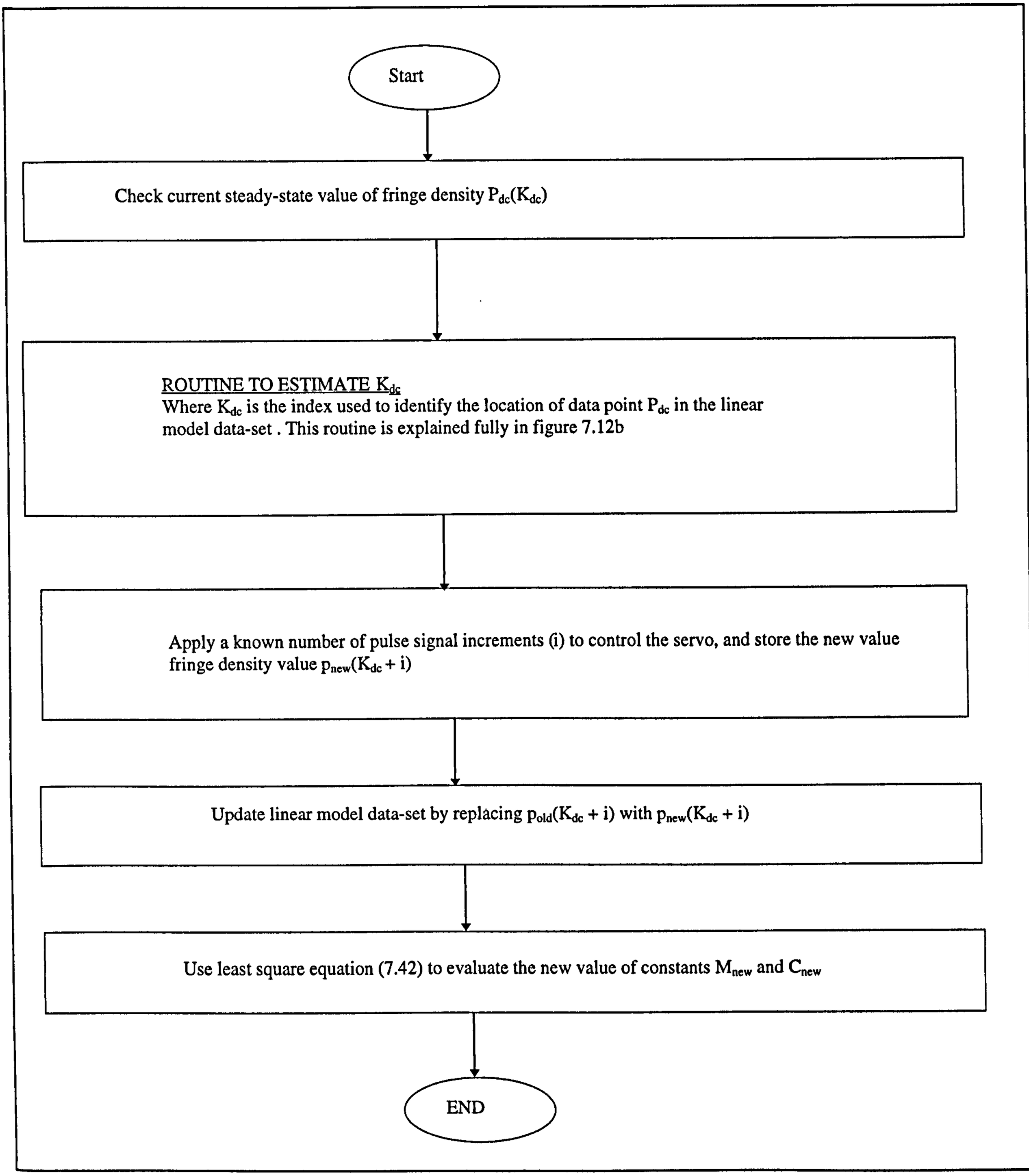


Figure 7.12a: Recursive least square algorithm to evaluate the dynamic constants of servo first-order model

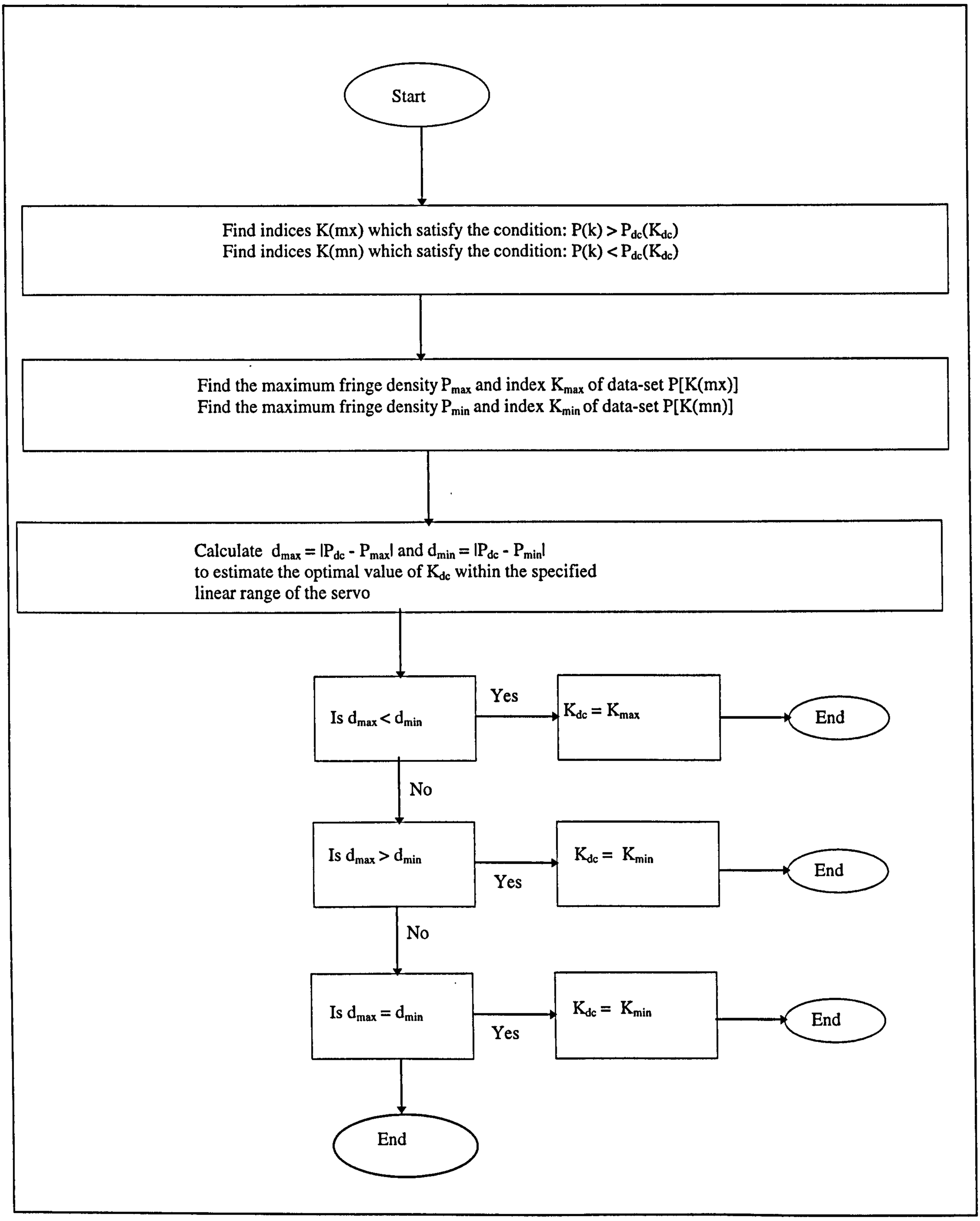


Figure 7.12b: Routine to estimate K_{dc}

In this work the non-linearity of the servo has been determined. Figures 7.13 and 7.14 shows the degree of backlash and stochastic noise present in the servo, for both the X-axis and Y-axis respectively.

The full characteristics of the servo can be described mathematically by modifying equation (7.46) to:

$$\mathbf{P}(s) = \mathbf{E_r}(s)\mathbf{G_b}(s)\,\mathbf{G_s}(s) + \mathbf{G_n}(s) + \mathbf{C} \tag{7.48}$$

$$\mathbf{P}(s) = \mathbf{G_b}(s)\mathbf{P_L}(s) + \mathbf{G_n}(s) \tag{7.49}$$

where, $\mathbf{P_L}(s)$ is fringe density of the servo after linearisation in the Laplace-domain, and is represented as:

$$\mathbf{P_L}(t) = \mathbf{M}\sum_{t=0}^{t=N}\mathbf{e_r}(t) + \mathbf{C} \tag{7.50}$$

in the time-domain

$\mathbf{M} = \begin{bmatrix} M_x \\ M_y \end{bmatrix}$: slope of $\mathbf{P_L}(t)$ for X and Y axis of servo

$\mathbf{C} = \begin{bmatrix} C_x \\ C_y \end{bmatrix}$: Y-intercept of $\mathbf{P_L}(t)$ for X and Y axis of servo

$\mathbf{G_b} = \begin{bmatrix} G_{bx} \\ G_{by} \end{bmatrix}$: Transfer function of backlash error in X and Y axis of servo

$\mathbf{G_n} = \begin{bmatrix} G_{nx} \\ G_{ny} \end{bmatrix}$: Transfer function of stochastic noise in X and Y axis of servo

the stochastic error is calculated as:

$$\mathbf{e_n}(t) = \mathrm{MAX}(|\mathbf{P}(t) - \mathbf{P_L}(t)|) \tag{7.51}$$

where, MAX: is a function that calculates the maximum value of an array
$\mathbf{P}(t)$: is the array of *original* fringe density values
$\mathbf{P_L}(t)$: is the array of *linear-fitted* fringe density values.

and the backlash error is calculated as the averaged difference between the *forward* and *backward* value of $\mathbf{p_L}(t)$:

$$\mathbf{e_b}(t) = \sum_{t=1}^{t=N} \left(\frac{\mathbf{p_L^+}(t) - \mathbf{p_L^-}(t)}{N} \right) \tag{7.52}$$

where, $\mathbf{p_L^+}(t)$: is the value of $\mathbf{p_L}(t)$ in the forward direction

$\mathbf{p_L^-}(t)$: is the value of $\mathbf{p_L}(t)$ in the backward direction

N: is the total number of data points

Backlash errors of 1.1 and 1.7 fringes were calculated using the data indicated in figures 7.13 and 7.14 respectively. Figures 7.15 and 7.16 show the magnitude of stochastic noise of the servo. The figures indicates a linear fit of the input/output data point and values of stochastic noise $e_n(t)$ of 0.75 and 1.72 fringes respectively.

Equation (7.50) is a refined model of the servo input/output characteristics and has been used to optimise the control of the interferometer (ie see section 7.7 for more details). As a result of the non-linear *time-varying* characteristic of the servo shown graphically in figures 7.17a and 7.17b respectively, values of **M** and **C** in equation (7.50) have to modified in-situ when controlling the interferometer. The figures indicate that the characteristic of the Y-axis servo is more prone to vary in time than the X-axis servo. Table 7.3 shows calculated values of **M** and **C** for 4 sets of input/output data of the X and Y-axis servo, indicating their respective time varying characteristics.

Linear constants	**X-axis servo**	**Y-axis servo**
M: Slope	-0.76 -0.73 -0.73 -0.73	-0.73 -0.61 -0.64 -0.56
C: Y-intercept	11.06 11.20 11.13 11.33	9.73 8.73 8.33 8.27

Table 7.3: Linear constants of servo indicating its time-varying characteristic

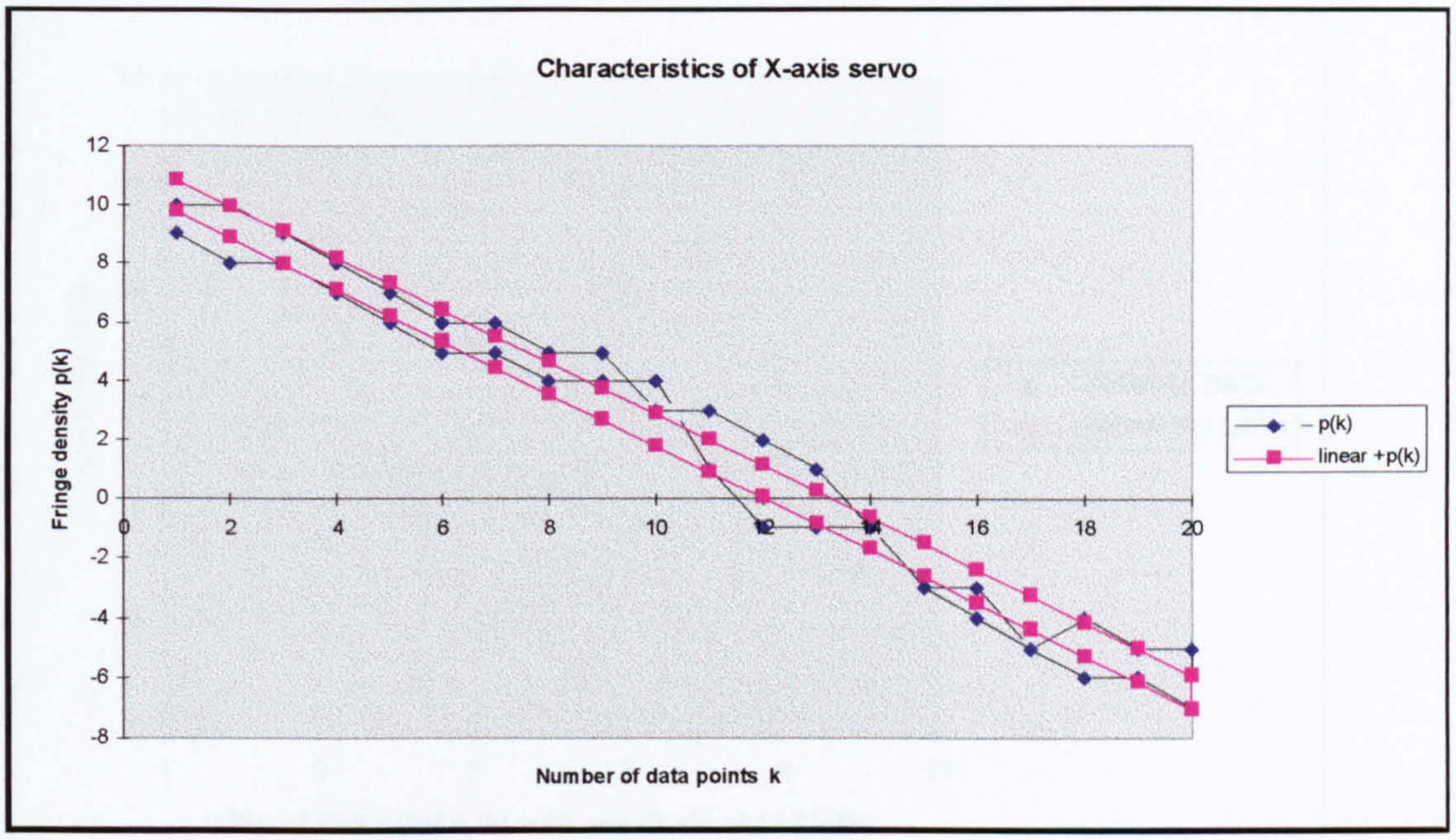


Figure 7.13: Measurement of backlash in X-axis servo

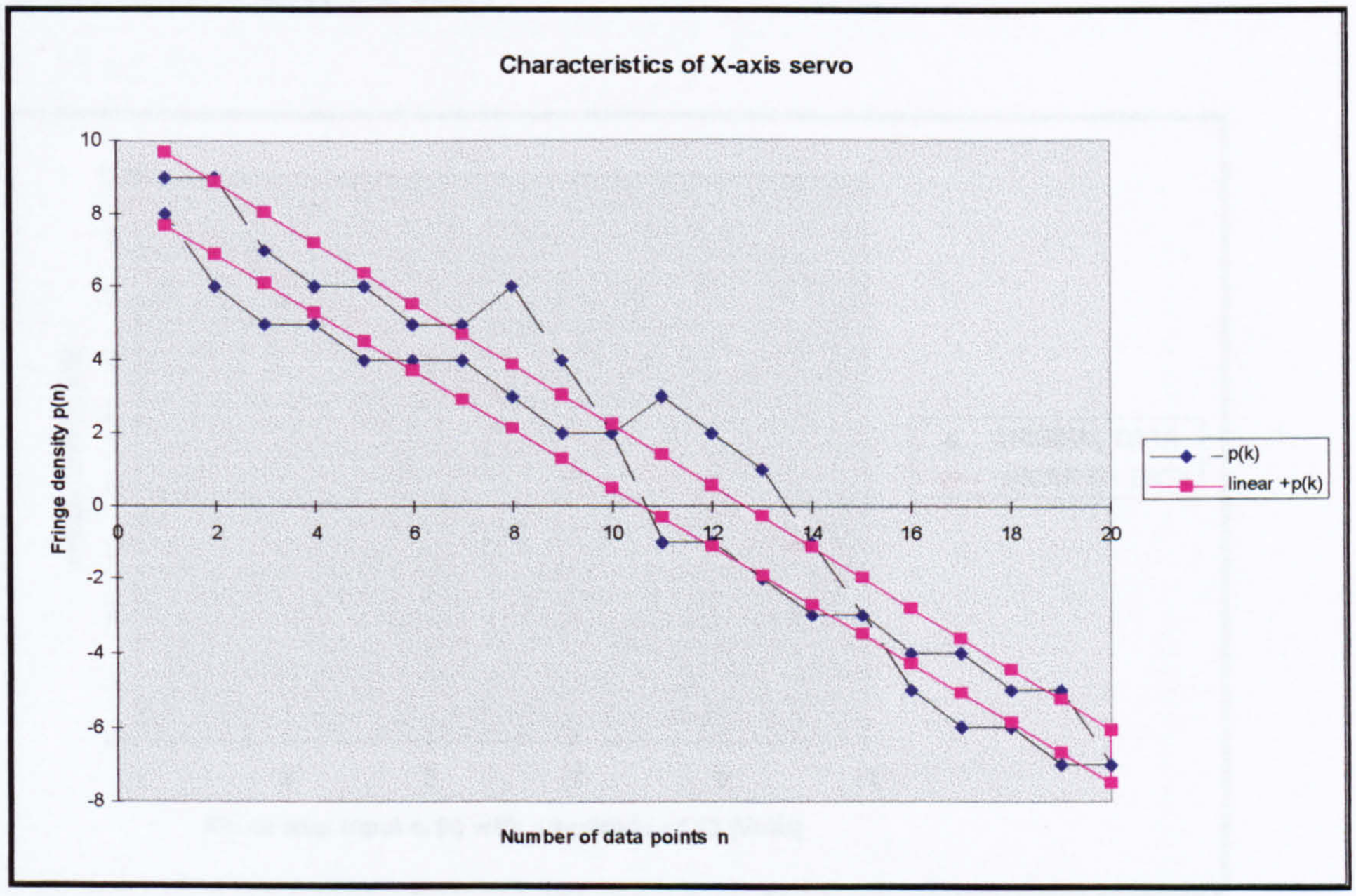


Figure 7.14: Measurement of backlash in Y-axis servo

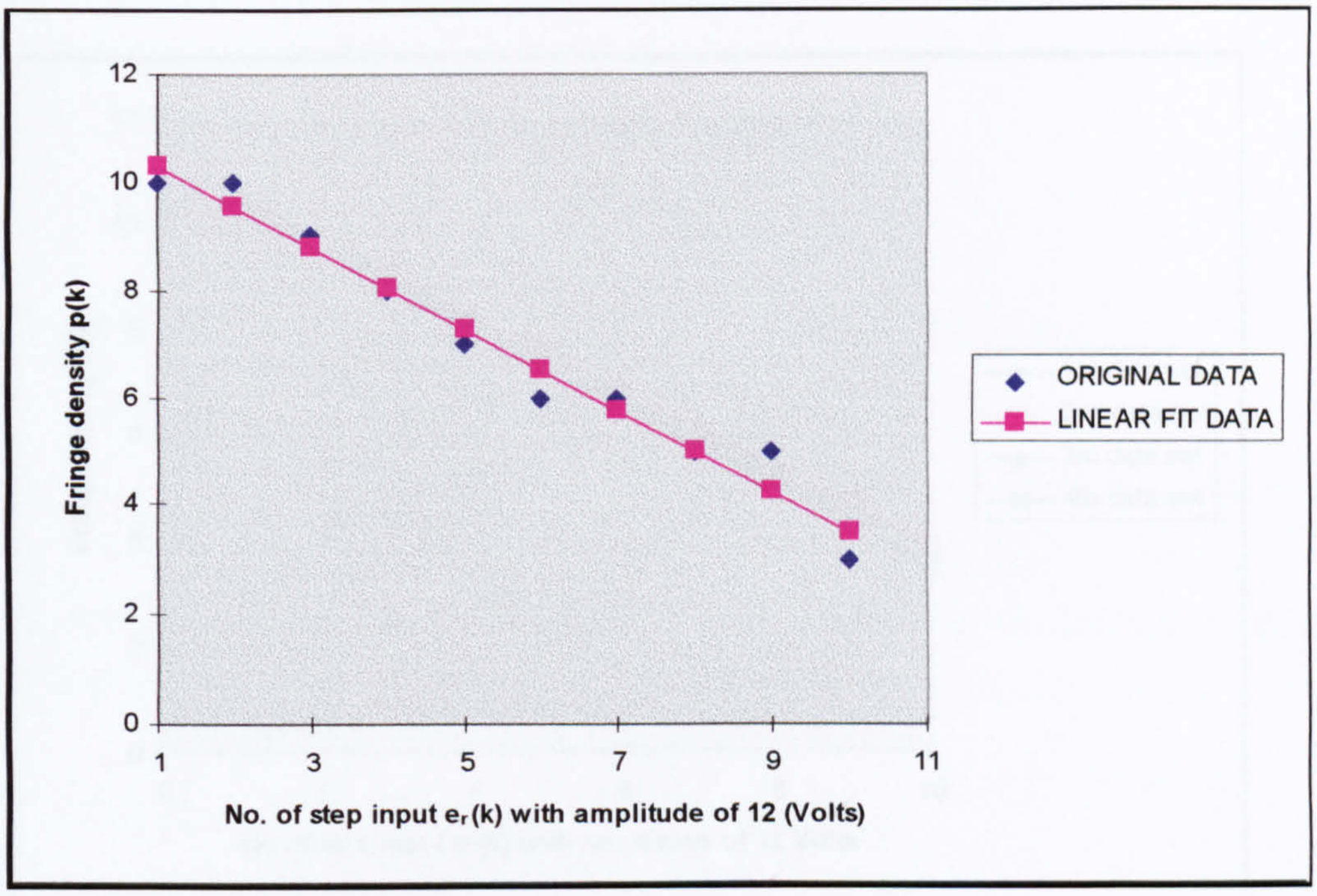


Figure 7.15: Measurement of stochastic noise in X-axis servo (measurements are in the forward direction only)

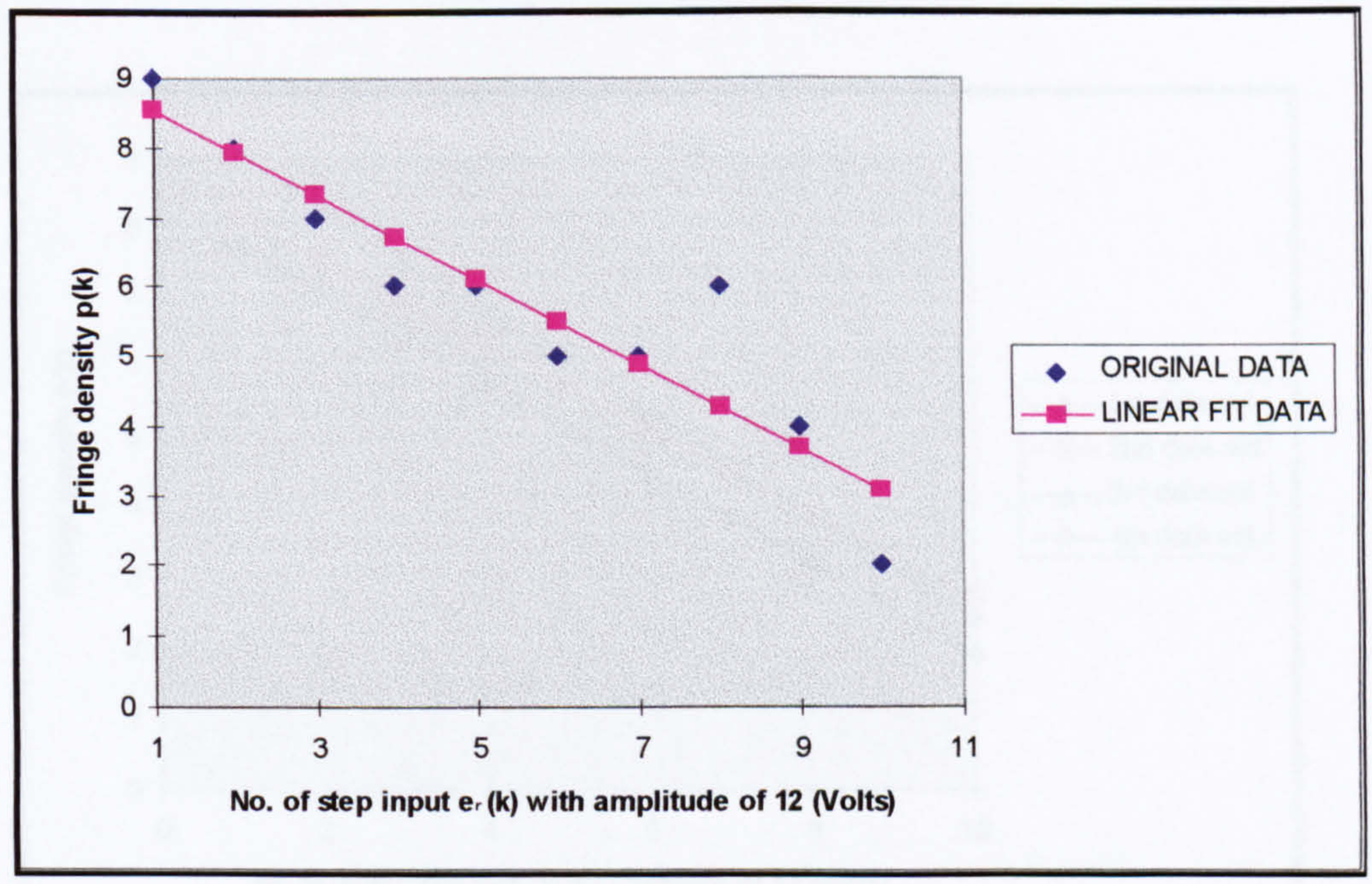


Figure 7.16: Measurement of stochastic noise in Y-axis servo (measurements are in forward direction only)

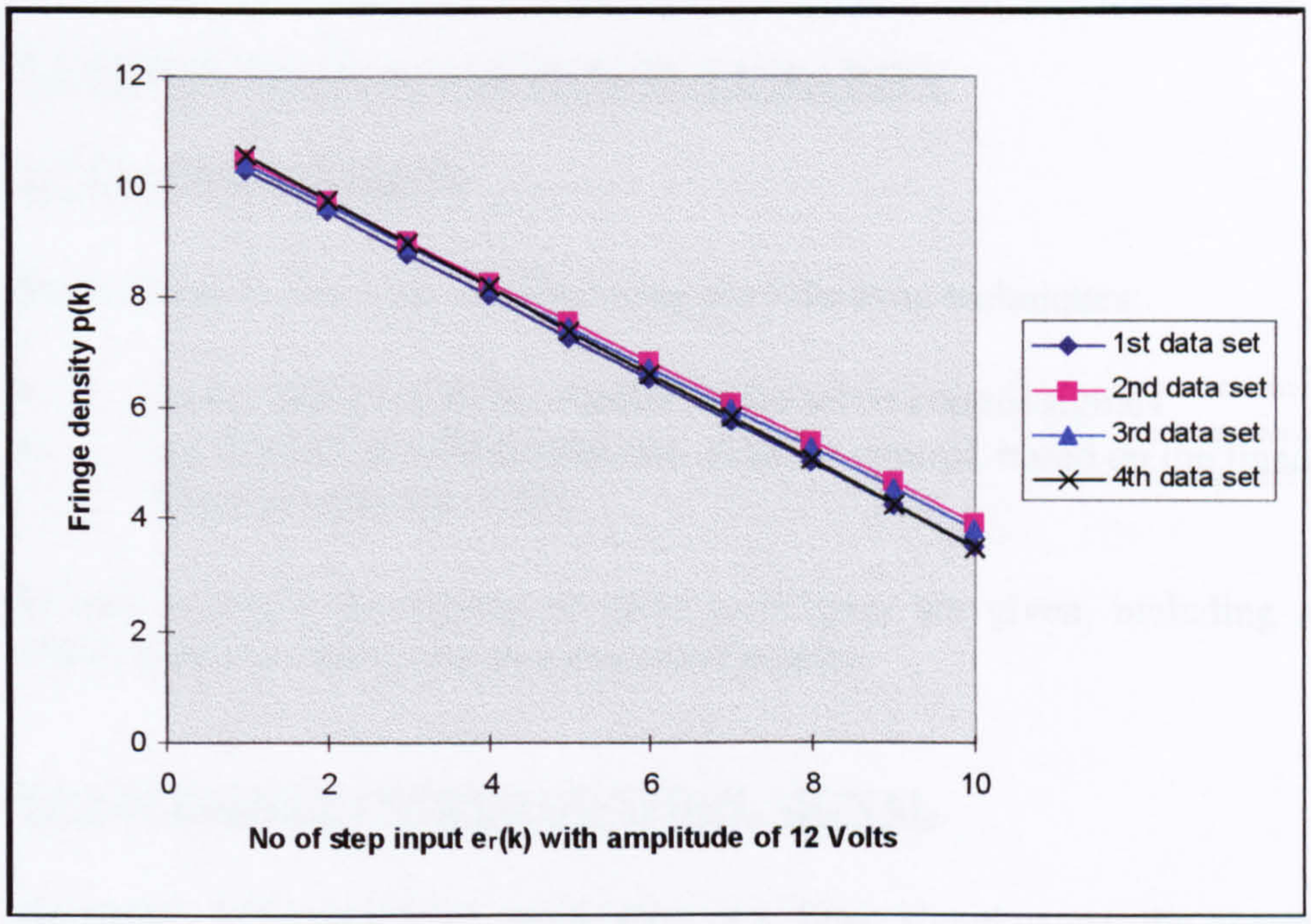


Figure 7.17a: Linear relationship between p(k) and $e_r(k)$ showing time-varying characteristic in X-axis servo (measurements are for k = 10 data points and in forward direction)

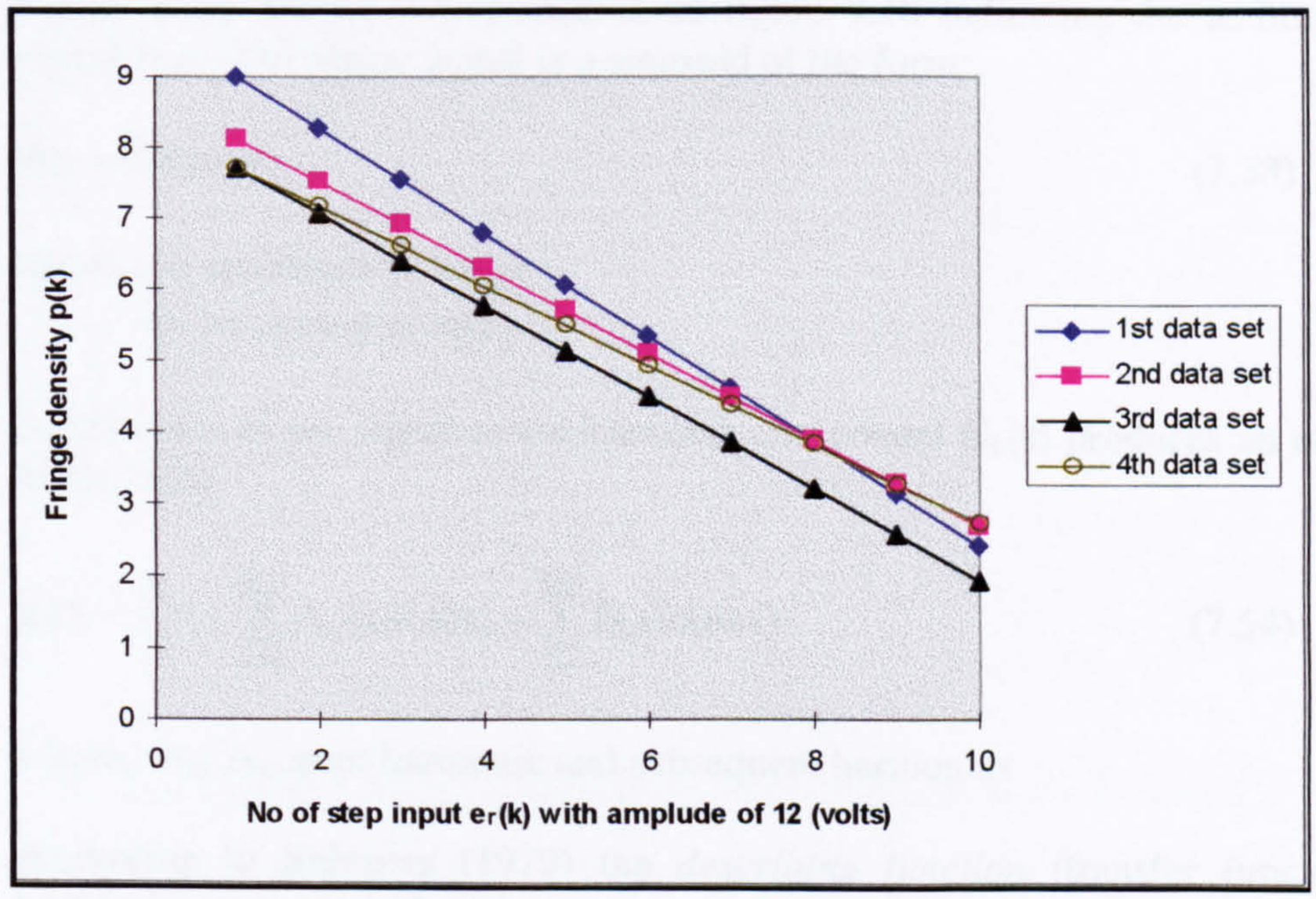


Figure 7.17b: Linearised relationship between p(k) and er(k) show time-varying characteristic in Y-axis servo (measurements are for k = 10 data points and in forward direction)

7.7 REDUCTION OF SERVO NON-LINEARITY

7.7.1 INTRODUCTION

Servo linearity was improved by using the following techniques:

- superposition of dither signals on the servo control signals
- application of model-reference adaptive control, based on the linear model given in equation (7.50)

In this section a description of these techniques are given, including some results which indicate a reduction in servo non-linearity.

7.7.2 DITHERING SERVO CONTROL SIGNAL

According to Cook (1994), application of a dither signal causes the operating point of the servo to sweep repeatedly over a certain range around its nominal position, on a time-scale much shorter than that of the system dynamics. This technique involves the superposition of dither signal (**d**(t)) (usually a sine or square waveform) on the servo control signal (**p**(t)). Generally the dither signal *averages* the non-linearity error.

Figure 7.18a shows a modification of figure 7.10 indicating the addition of dither signal D(s). The dither signal is a sinusoid of the form:

$$d(t) = \Phi\sin(\omega t) \qquad (7.53)$$

where, Φ: amplitude in fringes
ω: frequency in rad/s,

Application of the signal to the backlash component $G_b(s)$ produces an output signal of the form:

$$\chi(t) = \frac{A_o}{2} + \sum_{n=1}^{n=\infty} A_n\cos(n\omega t) + \sum_{n=1}^{n=\infty} B_n\sin(n\omega t) \qquad (7.54)$$

where, A_o, A_n: zero harmonic and subsequent harmonics

According to Shinners (1979) the *describing function* (transfer function) of the backlash component $G_b(s)$ is:

$$G_b(s) = \frac{\chi_1(s)e^{i\phi}}{\Phi} \qquad (7.55)$$

where, $\chi_1(t) = \sqrt{A_1^2 + B_1^2}$: modulus of the fundamental components of $\chi(k)$

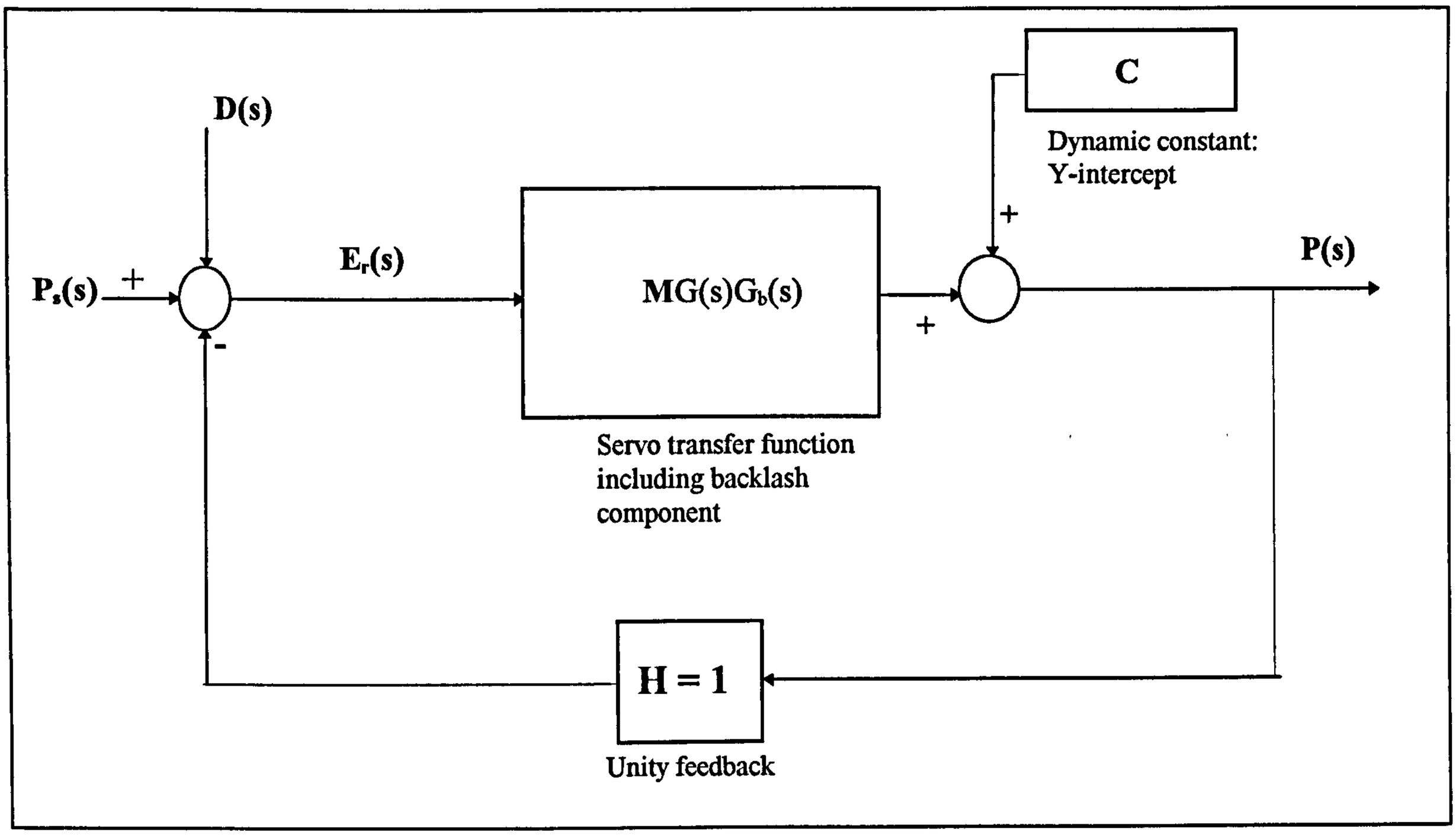


Figure 7.18a: System diagram of servo indicating backlash component $G_b(s)$

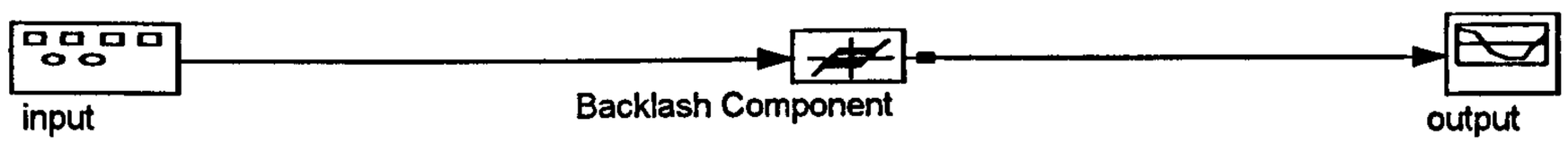


Figure 7.19: SIMULINK simulation diagram for servo dithering test

$\phi = \tan\left(\frac{A_1}{B_1}\right)$: is the phase shift

It is possible to reduce the backlash error $e_b(t)$ ideally to zero by choosing the appropriate amplitude (Φ) and frequency (ω) of dither signal (d(t)). The magnitude of the backlash error approaches zero when amplitude of d(t) is half the backlash error, and when the frequency of the dither signal is relatively high (when compared to the backlash error transients).

In this work a simulation package called SIMULINK developed by Mathworks* was used to analyse the response of $G_b(s)$ (with a theoretical backlash error, e_b = 1 fringe) at different values of Φ and ω of the dither signal. Figure 7.19 shows the simulation system diagram used in this exercise, indicating the dither signal input, backlash component and the output.

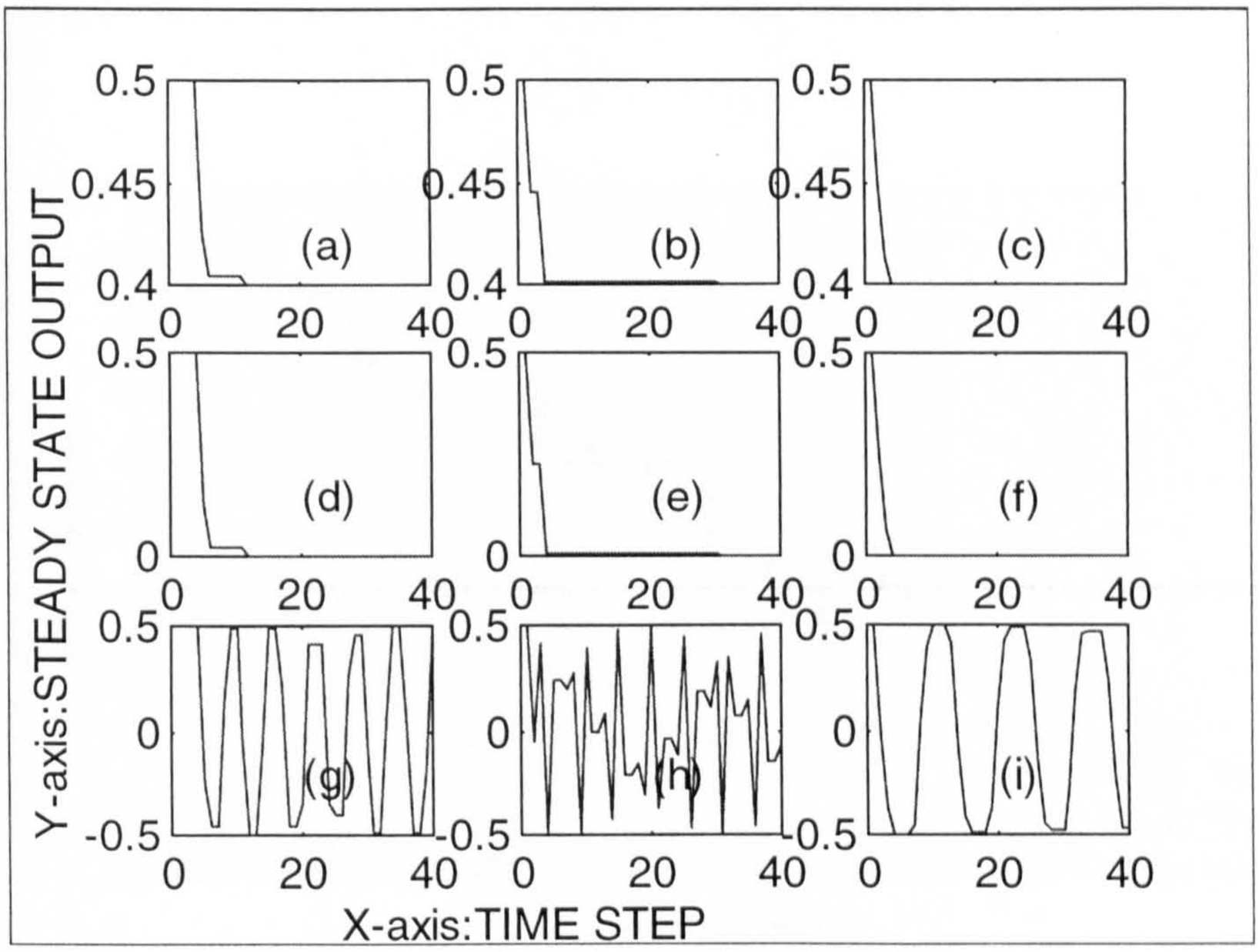
Simulation tests were carried out for Φ = 0.1, 0.5 and 1 fringe , and ω = 1, 10 and 100 rad/sec respectively. Output responses for the test are:

Φ = 0.1 at ω = 1, 10 and 10 is shown in figure 7.20a to 7.20c
Φ = 0.5 at ω = 1, 10 and 10 is shown in figure 7.20d to 7.20f
Φ = 1 at ω = 1, 10 and 10 is shown in figure 7.20g to 7.20i

Figures 7.20d to 7.20f confirm that the backlash error is zeroed when $\Phi = e_b/2$ and figure 7.20f in particular indicates that the response is much faster at higher dither frequencies.

The response of the dither input on servo non-linearity has been investigated for X and Y - axis servo. Figures 7.13 and 7.14 shows the input/output characteristic of X and Y-axis servo, the backlash error of both servos e_{bx} and e_{by} are 1.1 and 1.7 fringes respectively. Figure 7.21 shows that, dithering the X-axis servo with dither signal:

* MathWorks, Inc. Cochituate Place, 24 Prime Park Way, Natick, MA



Figures 7.20a to 7.20f: Simulated output of backlash component using dithering at different values of Φ and ω

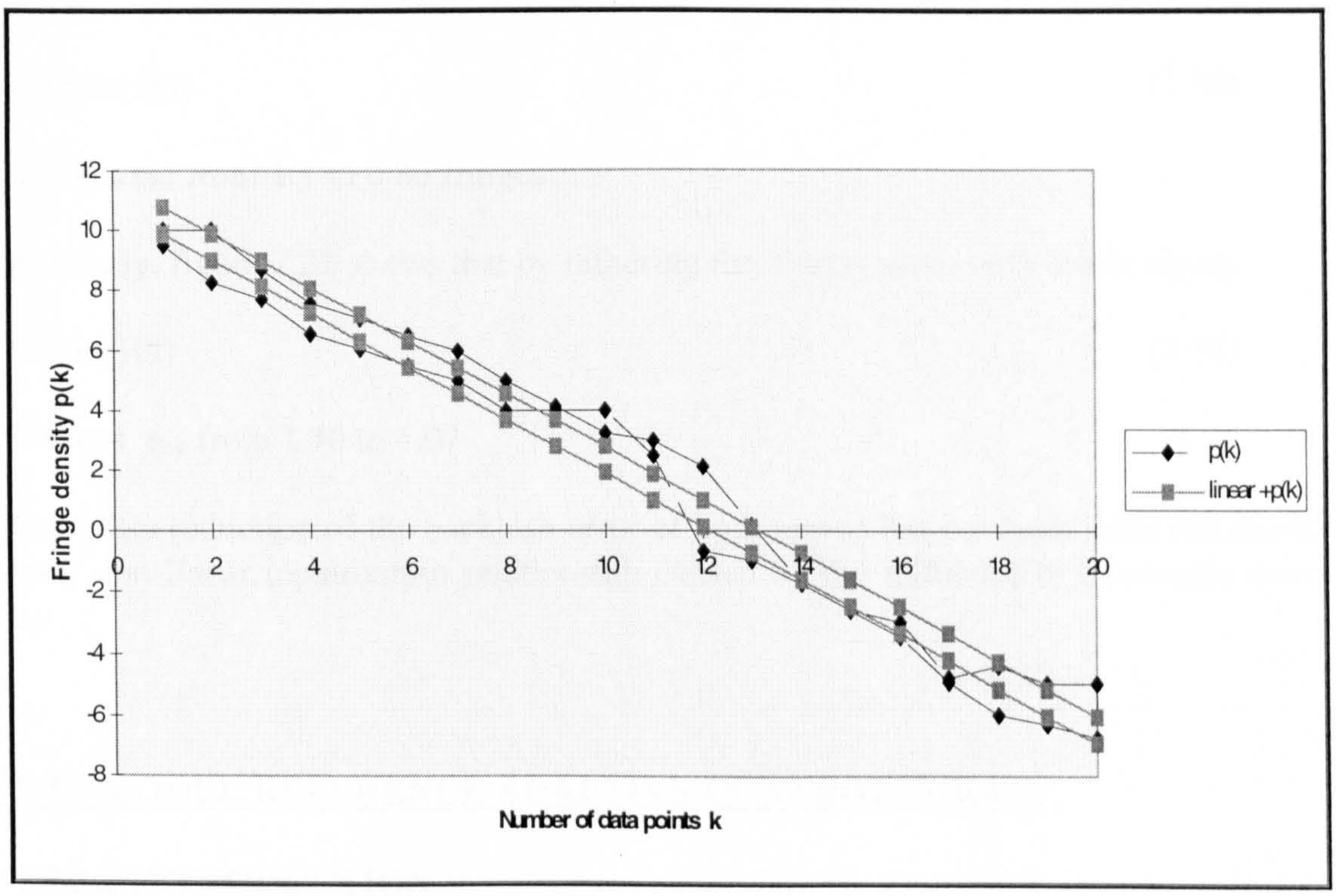


Figure 7.21: Improvement in backlash error of X-axis servo after dithering

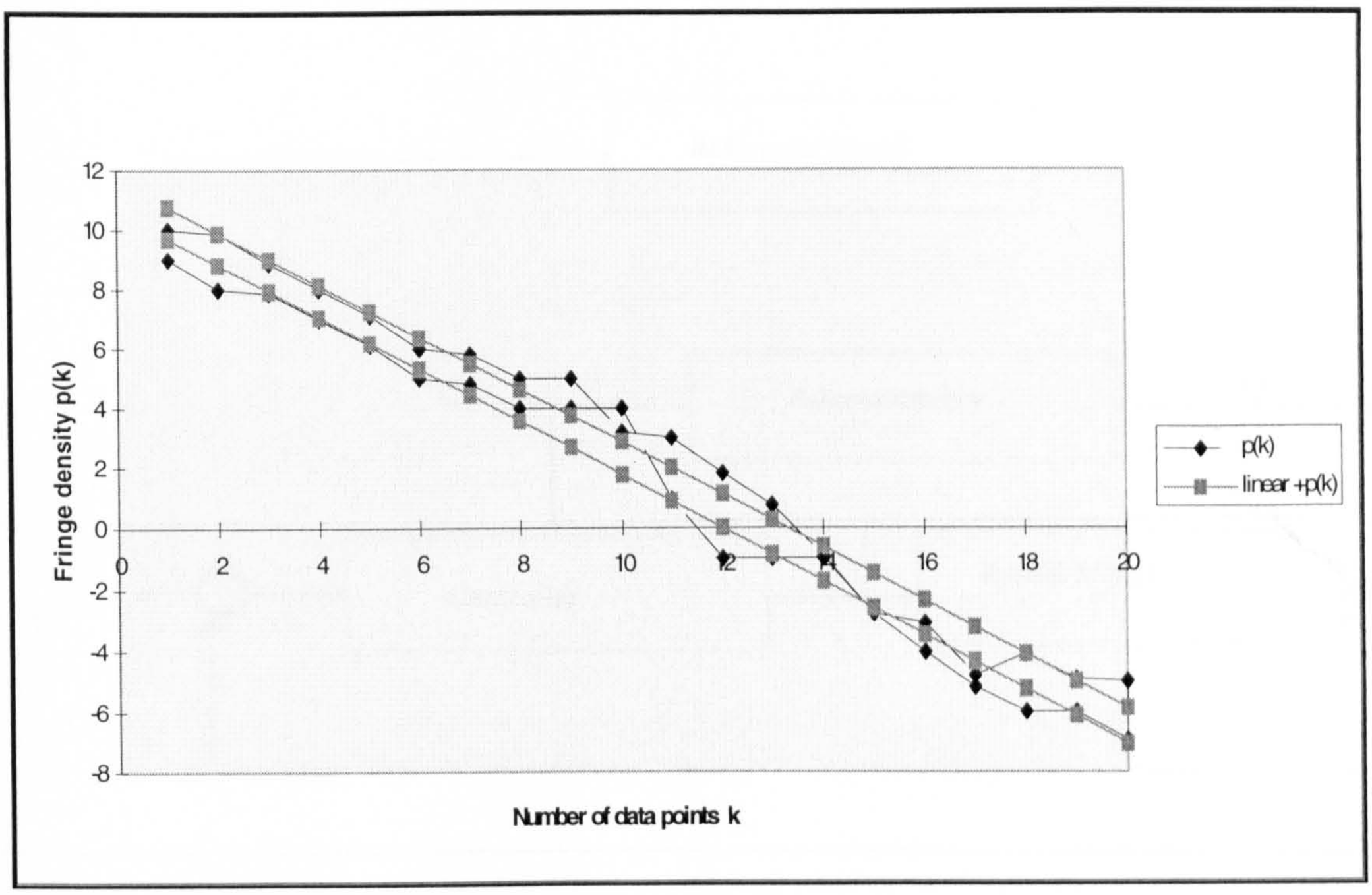


Figure 7.22: Improvement in backlash error of Y-axis servo after dithering

$0.55\sin(10t)$ (7.56)

reduces e_{bx} from 1.1 to 0.86 fringes

Similarly, figure 7.22 shows that by dithering the Y-axis servo with dither signal:

$0.8\sin(10t)$ (7.57)

reduces e_{by} from 1.70 to 1.07

Complete reduction of the backlash error of both servos has not been achieved due to their non-linear input/output relationship caused by the influence of stochastic noise $\mathbf{e_n}$.

7.7.3 MODEL REFERENCE ADAPTIVE CONTROL (MRAC)

7.7.3.1 INTRODUCTION

Model reference adaptive control (MRAC) is one of many adaptive control schemes used in applications as diverse as robotics, aircraft control, non-linear servomechanisms (Astrom & Wittenmark, 1995), large structural system control (Barkana & Kaufman, 1984), and precision machine tools (Slocum et al 1995). In this work MRAC is used to control the X and Y-axis servo such that its performance emulates a desired (servo) characteristic, usually referred to as a *reference model.*

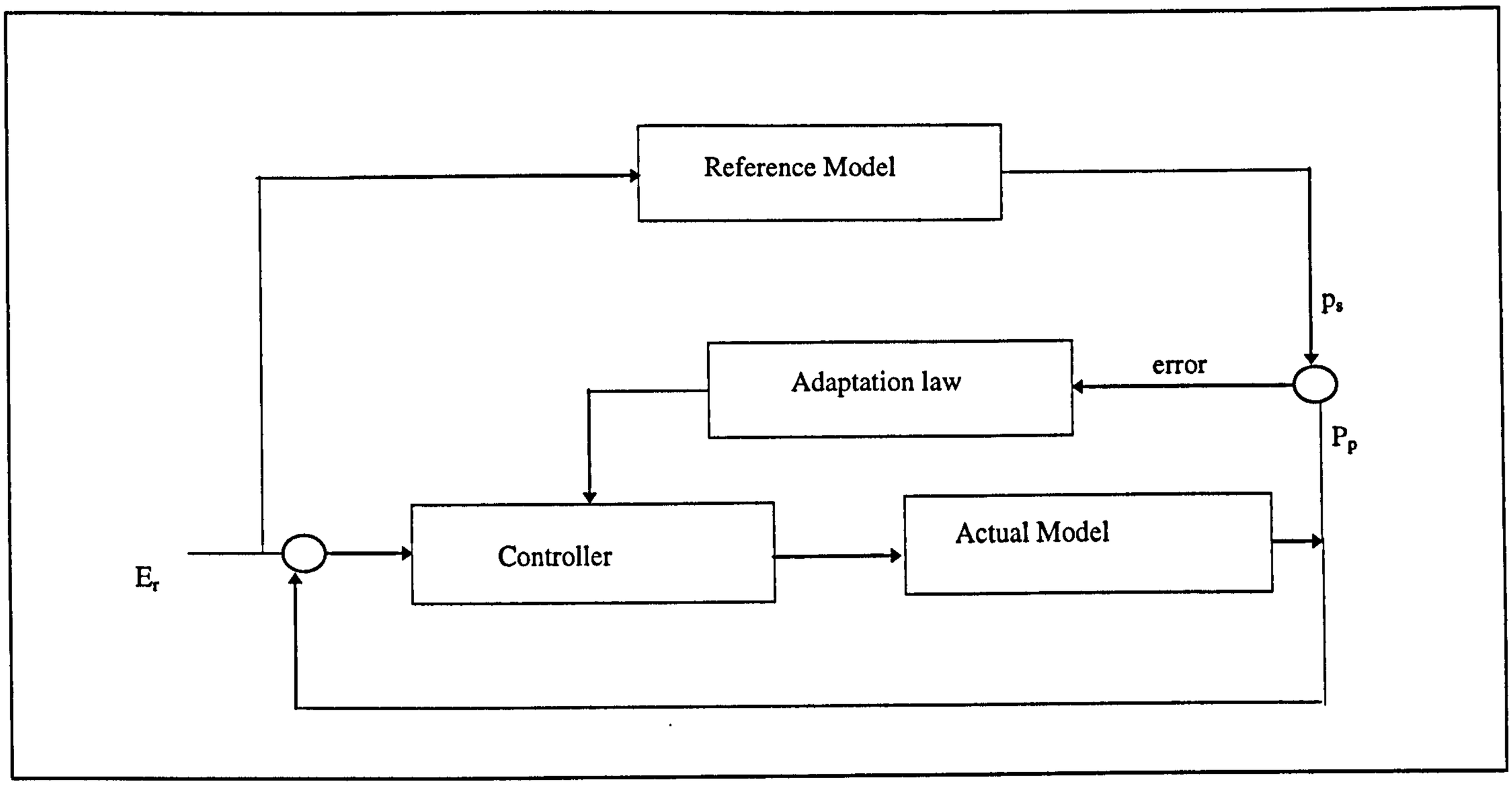


Figure 7.23: Principle of Model Reference adaptive control (after Isermann, 1992)

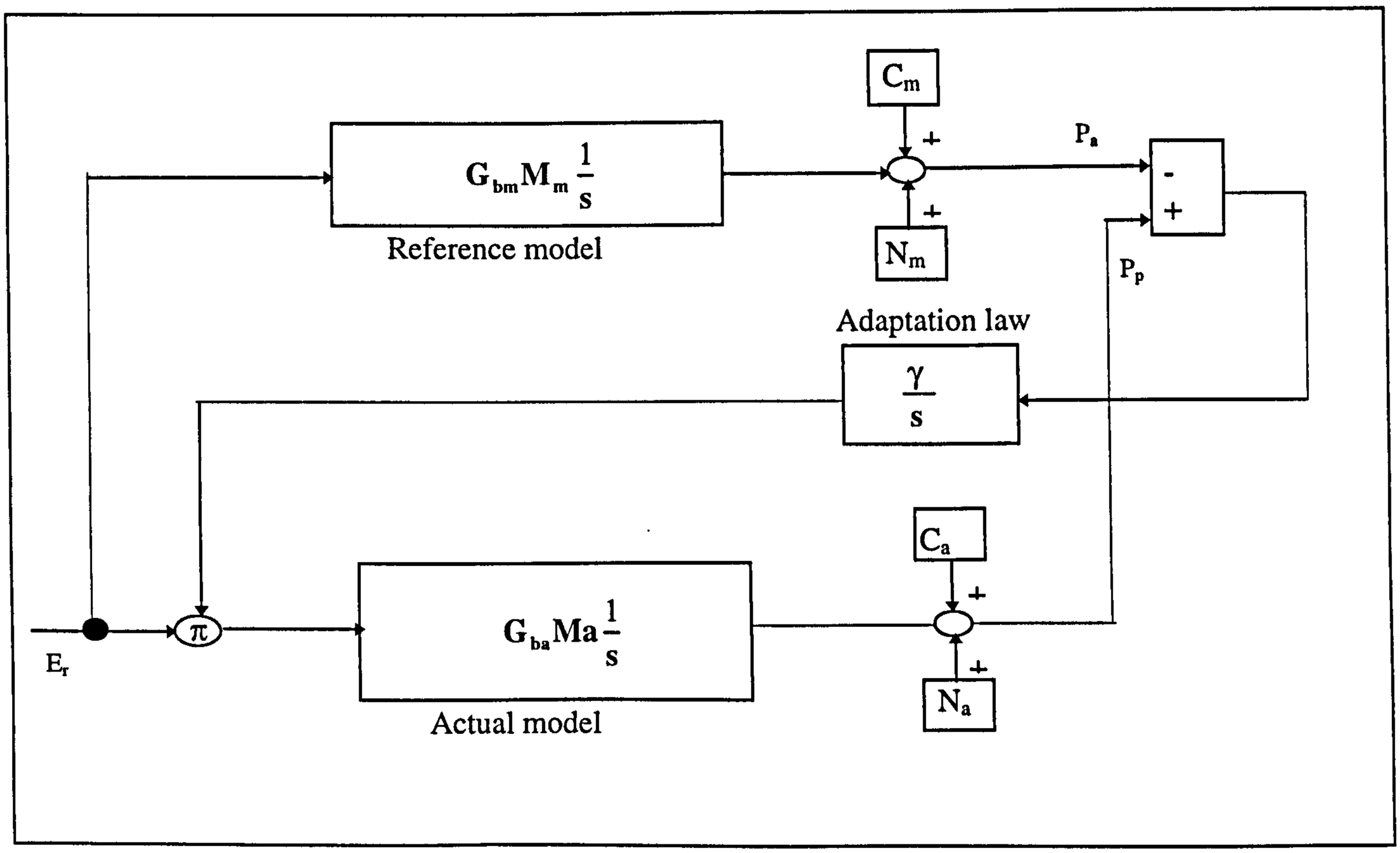


Figure 7.24: Schematic of the MRAC used in this project

The basic principle of MRAC is to compare the behaviour of the *controlled* servo with a reference (servo) model, and attempt to reduce the difference between them by adjusting control parameters (Cook, 1994). The techniques for adjusting the control parameters are *gradient methods* such as the MIT, Dressler and Price rules and *stability theory* such as Lyapunov and Monopoli theories (Hang & Parks, 1973). In designing the MRAC it is important that parameters of the controlled servo converges asymptotically with those of the reference model.

Figure 7.25 shows a schematic of a MRAC (Isermann et al, 1992). It has two loops, known as the *inner* and *outer* loops. The inner loop is an ordinary feedback loop which links the servo and controller. While the outer loop is used to adjust the controller parameters in order to minimise the error (ie the difference between the servo *physical* output p_p and the *model reference* output p_m).

An MIT-rule MRAC has been developed for the servo, simulation tests have been used to validate the adaptive controller. Description of the MRAC and its performance are provided in this section.

7.7.3.2 DEVELOPMENT OF SERVO MRAC

Figure 7.26 shows a schematic of the servo MRAC controller, adjustment of the *adaptation parameter* (η) is achieved by minimising the error (e) using the MIT rule. The MIT rule was developed by Osburn and Whitaker (1961) at MIT. The rule is based on adjusting η in such a way that half the squared of e:

$$\frac{e^2}{2} = \frac{1}{2}(p_p - p_m)^2 \qquad (7.58)$$

known as the *loss-function* $J(\eta)$ is minimised (Astrom & Wittenmark, 1995).

To minimise $J(\eta)$ the controller parameters must be changed in the direction of its negative gradient:

$$\frac{d\eta}{dt} = \gamma\frac{\partial J}{\partial \eta} = -\gamma\frac{\partial J}{\partial e}\frac{\partial e}{\partial \eta} = -\gamma e\frac{\partial e}{\partial \eta} \qquad (7.59)$$

where, γ: adaptation gain.

Equation (7.59) is known as the MIT adaptation rule. This equation is used to control the servo in order to achieve the following conditions:

- *physical* model is made to emulate the *reference* model: $G_m(s) = G_a(s)$
- e tends to zero: $e = p_p - p_m \rightarrow 0$

The actual ($G_a(s)$) and reference ($G_m(s)$) models are linear models determined by recursive least square method described in section 7.5.

In figure 7.24 $e = p_p - p_m = M_s G(s)\eta e_r - M_a G(s) e_r$ (7.60)

and the partial derivative of equation (7.56) is:

$$\frac{\partial e}{\partial \eta} = M_s G(s) e_r = \frac{M_a}{M_m} p_m \tag{7.61}$$

Hence the rate of change of the adaptation parameter is:

$$\frac{d\eta}{dt} = -\gamma \frac{\partial e}{\partial \eta} = -\gamma \frac{M_a}{M_m} p_m e = \gamma' p_m e \tag{7.62}$$

where, p_a, p_p: Output of the actual and reference servo models

M_s, M_P: Slope of the actual and reference servo models

G(s): integrator component in actual and reference servo models

$\gamma' = \gamma \frac{M_a}{M_m}$: modified adaptive gain

7.7.3.4 DETERMINATION OF ADAPTATION GAIN

The adaptation gain value γ essentially determines how the physical output p_p approaches the model reference output p_m. This parameter can be derived from:

$$s + \gamma p_m^0 e_r^0 M_p G(s) = s + \varpi G(s) = 0 \tag{7.63}$$

where, $p_m^0 = e_r^0 M_m$: reference model output (fringe density) at steady-state

e_r^0 : step input (voltage) at steady state

$G(s) = \frac{1}{s}$: adaptation integrator

$\varpi = \gamma p_m^0 e_r^0 M_p = \gamma (e_r^0)^2 M_a M_p$: adaptation constant (7.64)

Equation (7.64) is a linear equation of the servo related to its stability (Astrom & Wittenmark). Substituting the value of G(s), equation (7.63) can be modified to:

$$s^2 + \varpi = 0 \tag{7.65}$$

The adaptively controlled servo is stable when equation (7.60) has its poles in the left half-plane of a root locus diagram (Franklin et al, 1994). This occurs when

$$\varpi < 0 \tag{7.66}$$

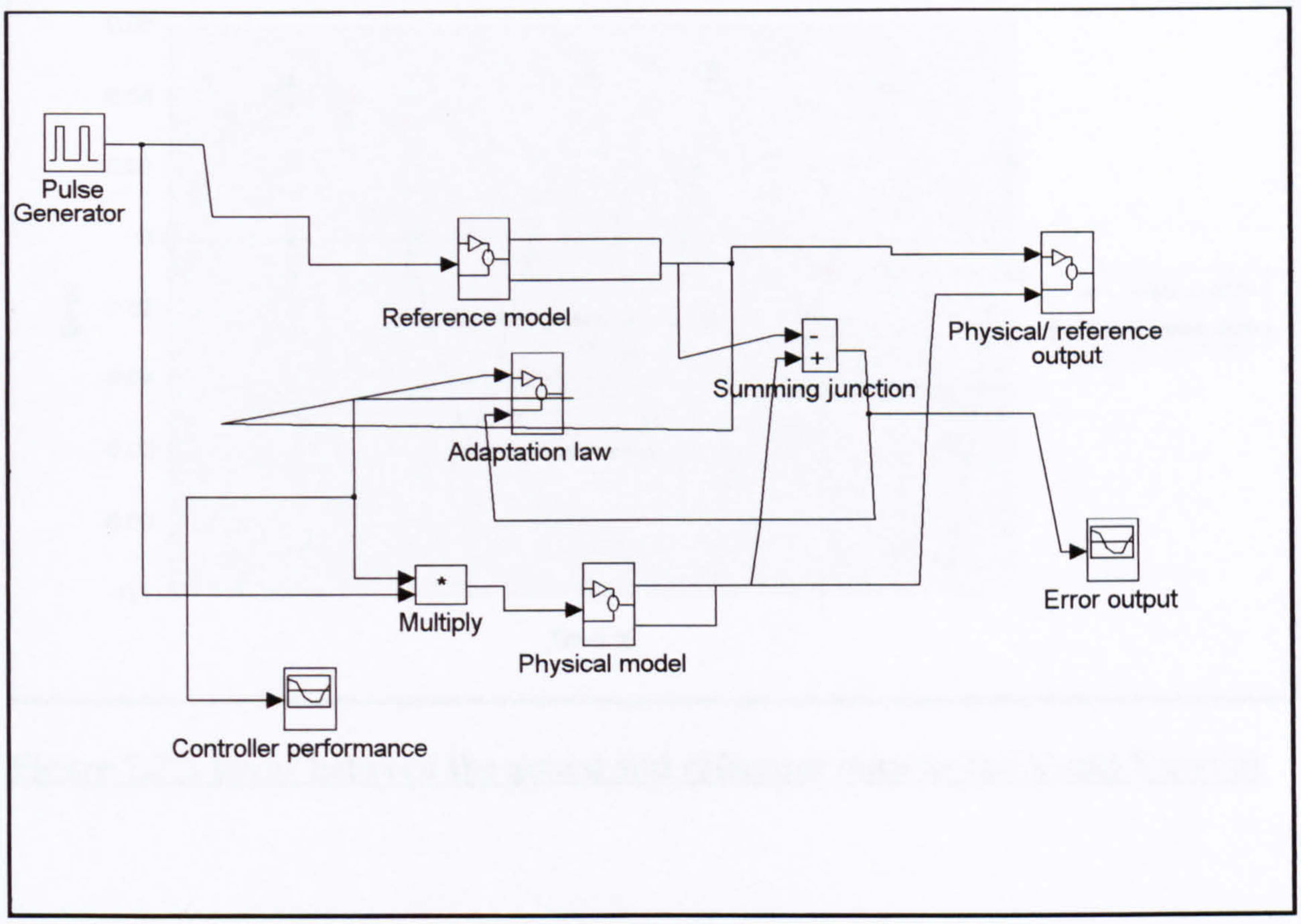


Figure 7.25: SIMULINK system diagram used to simulated the MRAC performance

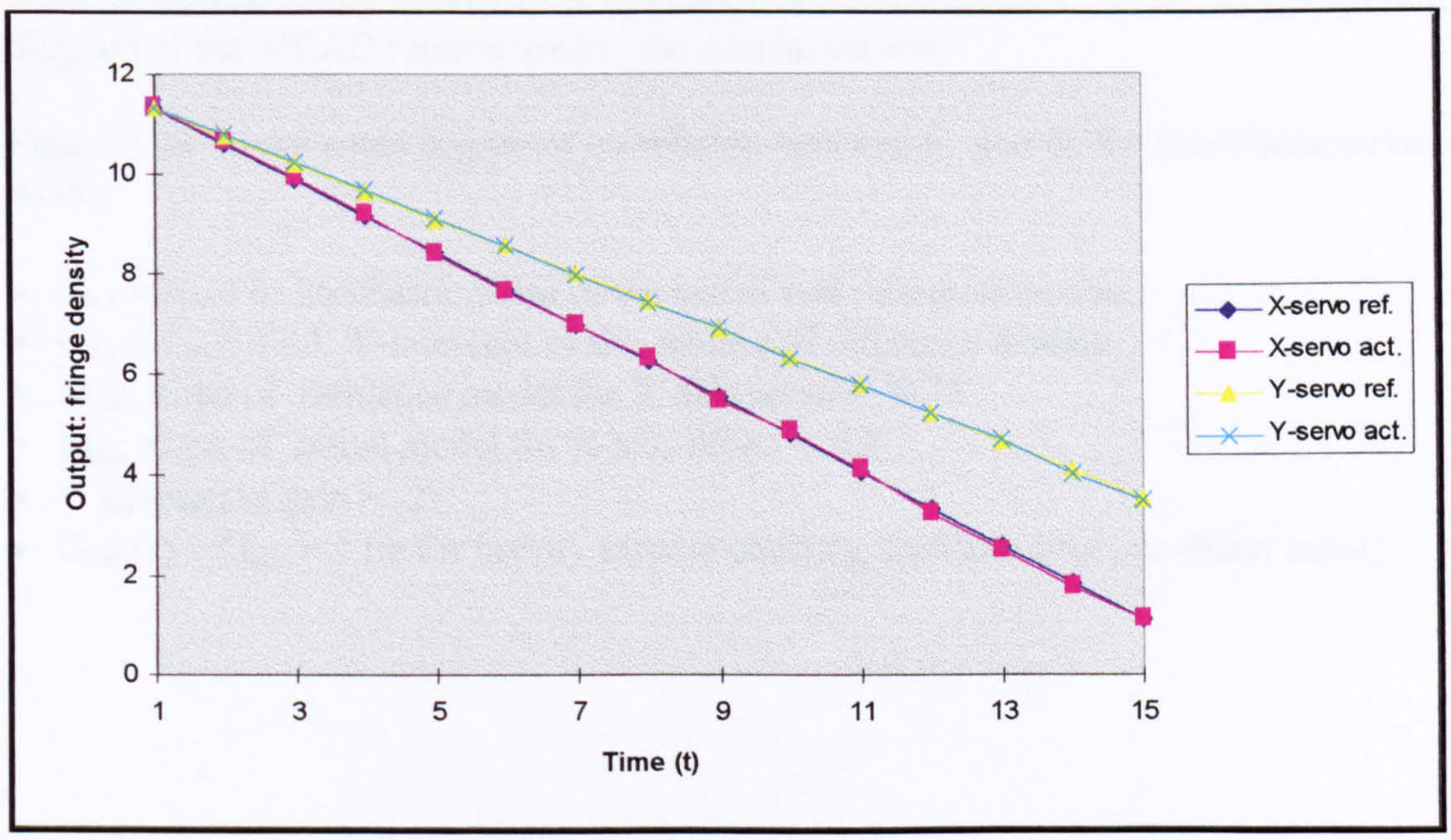


Figure 7.26: Correlation between the actual and reference models for X and Y servos

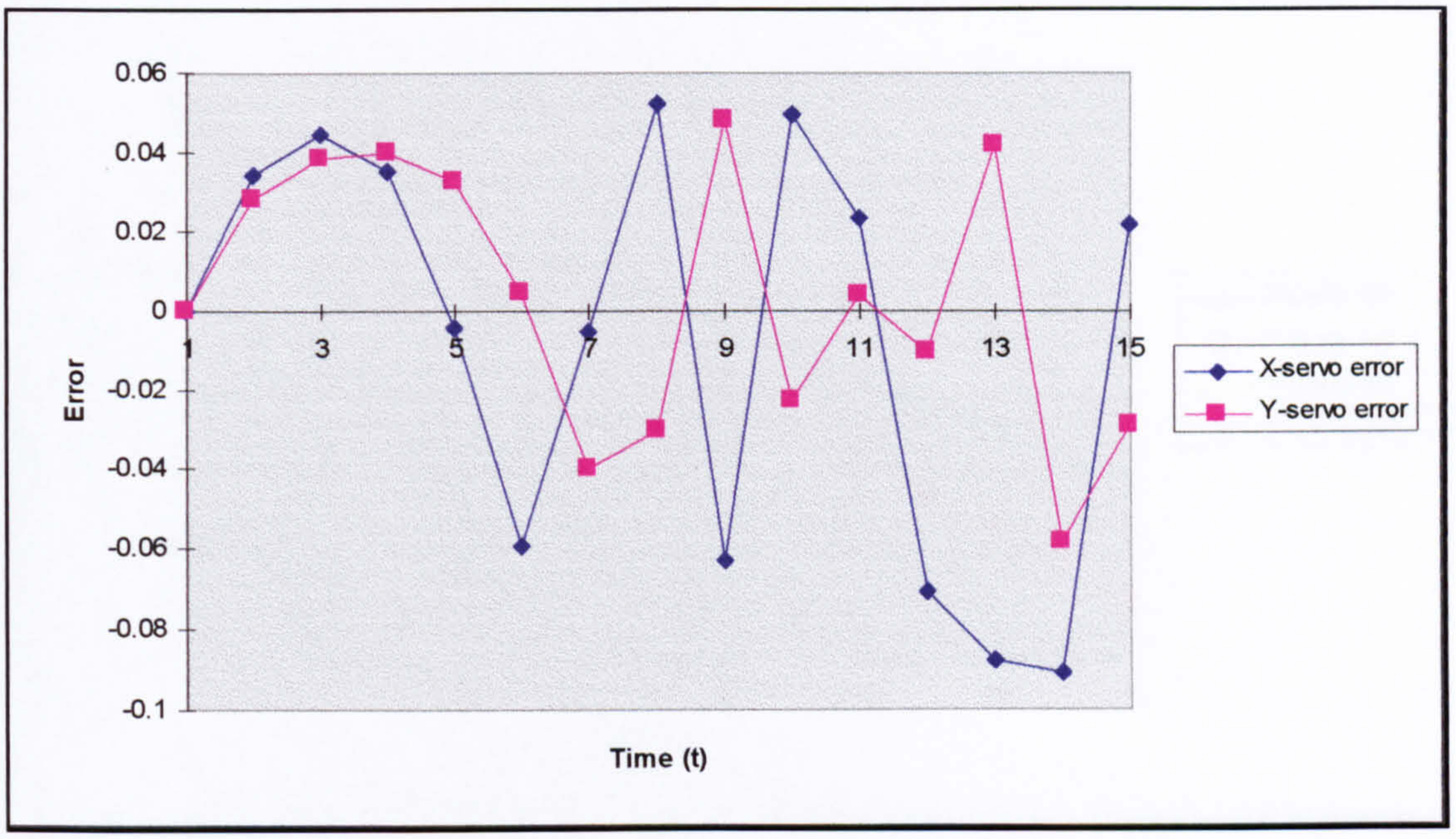


Figure 7.27: Error between the actual and reference outputs for X and Y servos

7.7.3.3 PERFORMANCE OF MRAC

Simulation tests of the X and Y axis servo using the MIT adaptation law confirmed that the *actual* output (P_a) approached the *reference model* output (p_m). These tests were performed using SIMULINK simulation software, figure 7.25 shows the system diagram of the MRAC routine used in the simulation test.

Figure 7.26 shows some degree of correlation between p_a and p_m for the X-axis servo, when:

- $N_{mx} = N_{ax} = 0$: Stochastic noise of the actual and reference models
- $C_{mx} = C_{ax} = 11.3$: Y-intercept of the actual and reference models
- M_{mx}: slope of reference model for X-axis servo = 0.73
- M_{ax}: slope of actual model for X-axis servo = 0.8
- γ: adaptation gain = -2
- $G_{bmx}(s) = G_{bpx} = 1$ (ie for brevity assume constant backlash error, no dither input)

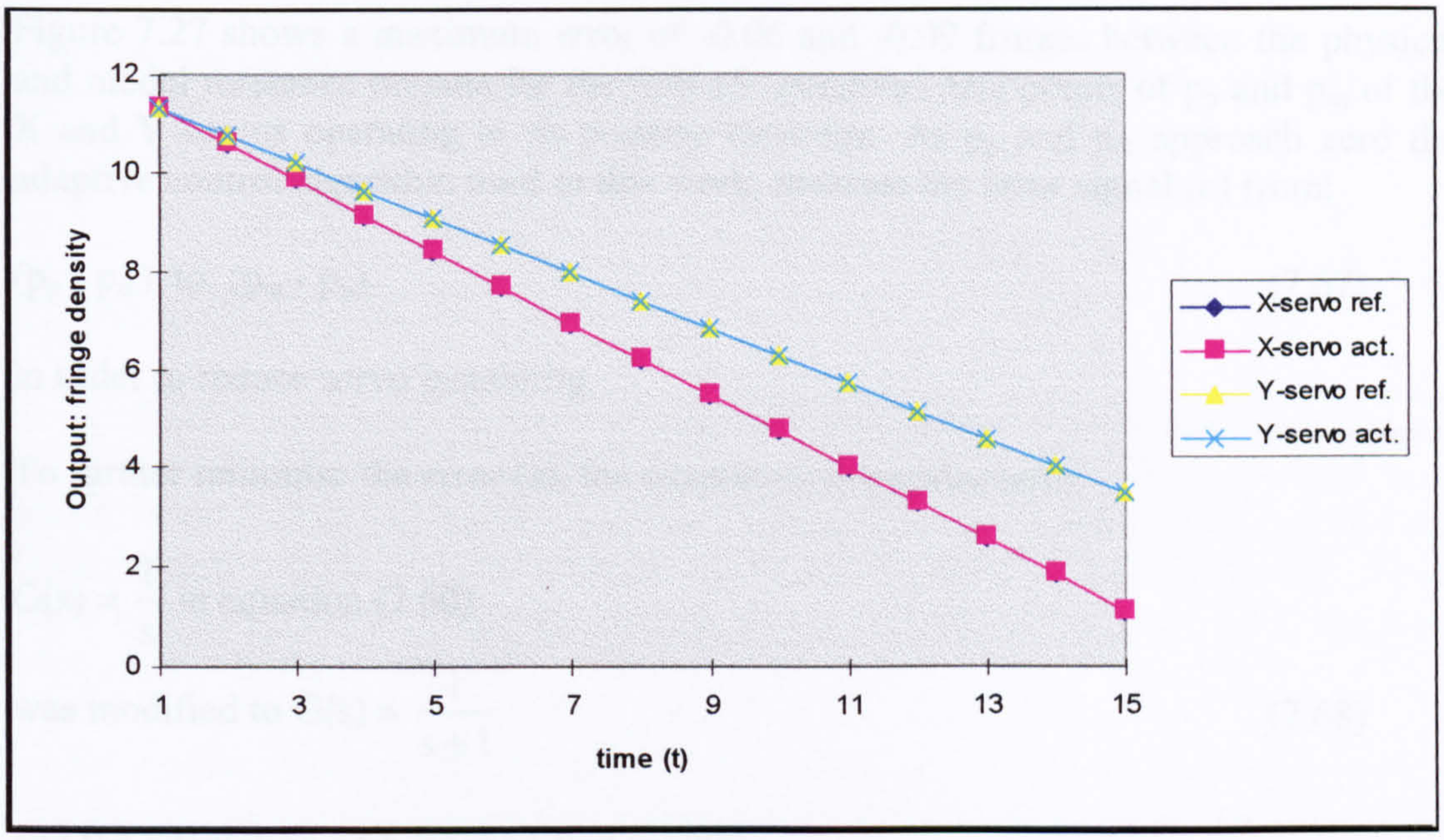


Figure 7.28: Improved correlation between the actual and reference models for servos

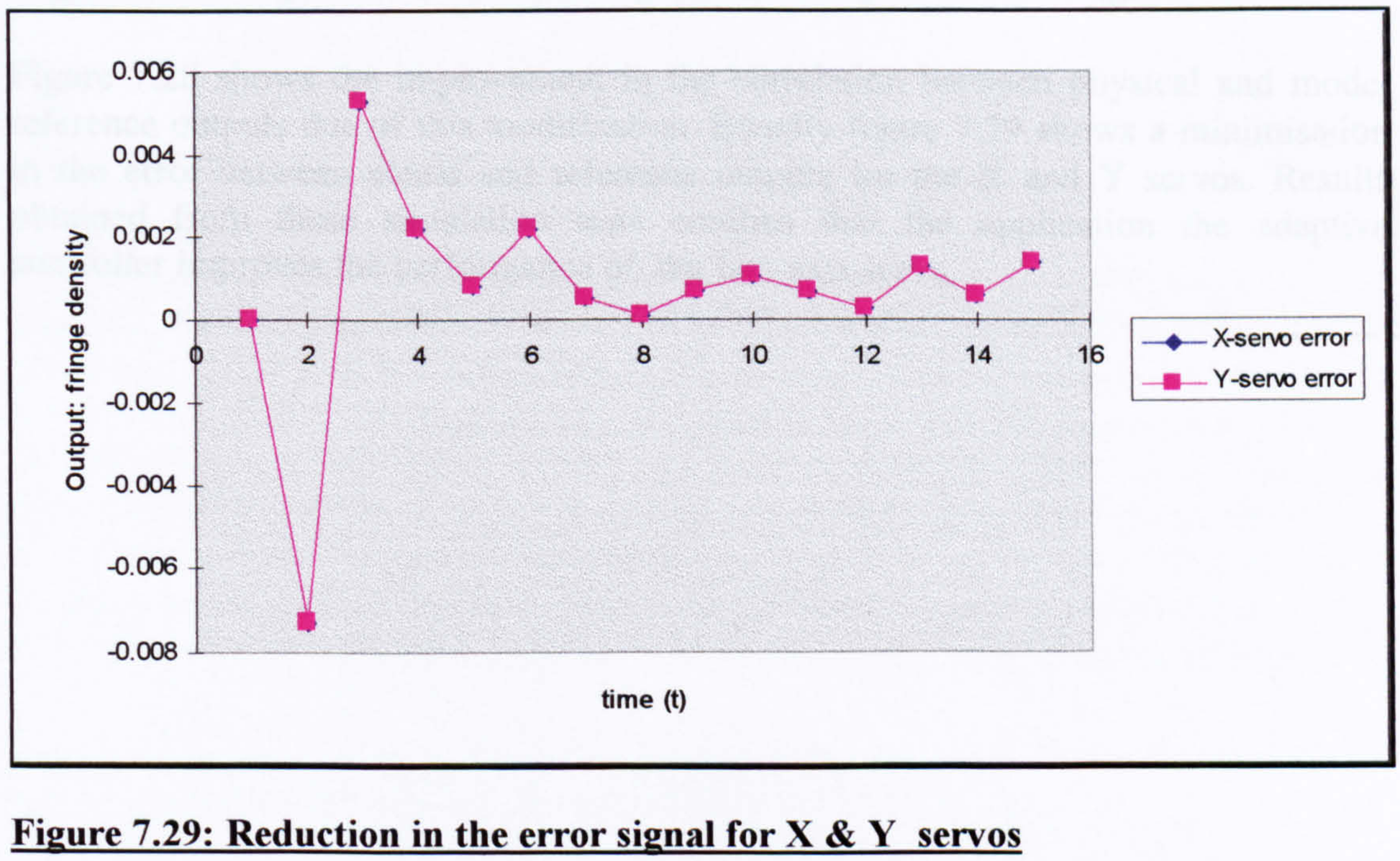


Figure 7.29: Reduction in the error signal for X & Y servos

Figure 7.27 shows a maximum error of -0.06 and -0.09 fringes between the physical and model reference outputs for the first 15 (positive) data points of p_p and p_m of the X and Y servos operating in its positive direction. As p_p and p_m approach zero the adaptive control algorithm used in this work, reverses the error signal (e) from:

$$(p_p - p_m) \text{ to } (p_m - p_p) \tag{7.67}$$

in order to reduce servo instability.

To further minimise the error (e), the adaptation integrator term:

$G(s) = \frac{1}{s}$ in equation (7.60)

was modified to $G(s) = \frac{1}{s+1}$ (7.68)

and the adaptation gain (γ) modified to -200.

The numerator of $\frac{1}{s+1}$, (whose root is sometimes referred to as a *pole*), influences the decay rate of the error signal. It will contribute an exponential term e^{-t}, to the original error output of e^0; thus reducing the instability of the error signal.

Figure 7.28 shows the improvement in the correlation between physical and model reference outputs due of this modification. Equally figure 7.29 shows a minimisation in the error between actual and reference outputs for the X and Y servos. Results obtained from these simulation tests confirm that the application the adaptive controller improves the performance of the two-axis servo.

CHAPTER 8: MEASUREMENT OF SPINDLE ERROR MOTIONS

8.1 INTRODUCTION

As described in section 7.2, the interferometer used in this work was adjusted with a aid of the two-axis servo, interferograms of nominally constant fringe density were acquired, processed and stored at chosen angular intervals of the spindle revolutions, typically this process was repeated to obtain an averaged measurement. The stored interferograms were subsequently used to provide spindle metrology data in the form of axial and tilt error motions.

In this chapter a description of spindle error motions (axial and tilt) measurements and techniques for reducing the noise present in the error motion signal are given. The spindle error motions can be calculated using either the interferogram spatial or Fourier transform data, both of these approaches are described in this chapter. Also interpretation of the *asynchronous* and *average* components of the error motions presented in *polar* form are illustrated. The software developed for measuring the spindle run-out are outlined in the program listings booklet (accompanied with the thesis) in section 5.

8.2 ACQUISITION OF SPINDLE METROLOGY DATA

The spindle axial (d_z) and tilt (E_x and E_y) error motions are measured using the acquired values of interferogram orientation, fringe spatial frequency, fringe spatial phase and fringe peak position. Prior to this measurement the rotating interferograms are checked for the condition of constant spatial frequency (or fringe density) using the interferometer adjustment routine described in figure 8.1. This is followed by a data acquisition routine used for sampling, storing and pre-processing interferograms of constant spatial frequency described in figure 8.2. A total of 30 data-points (interferograms) were sampled as the spindle revolved at approximately 60 rpm. This measurement was repeated 4 times in order to obtain the *average* error motion data.

8.3 MEASUREMENT OF AXIAL ERROR MOTIONS

8.3.1 INTRODUCTION

Spindle axial error motion can be measured by using either the fringe spatial phase evaluated in the spatial domain (interferogram in X-Y space) or by using the fringe spatial phase obtained in the Fourier domain (interferogram phase component of FFT transform in U-V space). The following sub-sections describe these measurements.

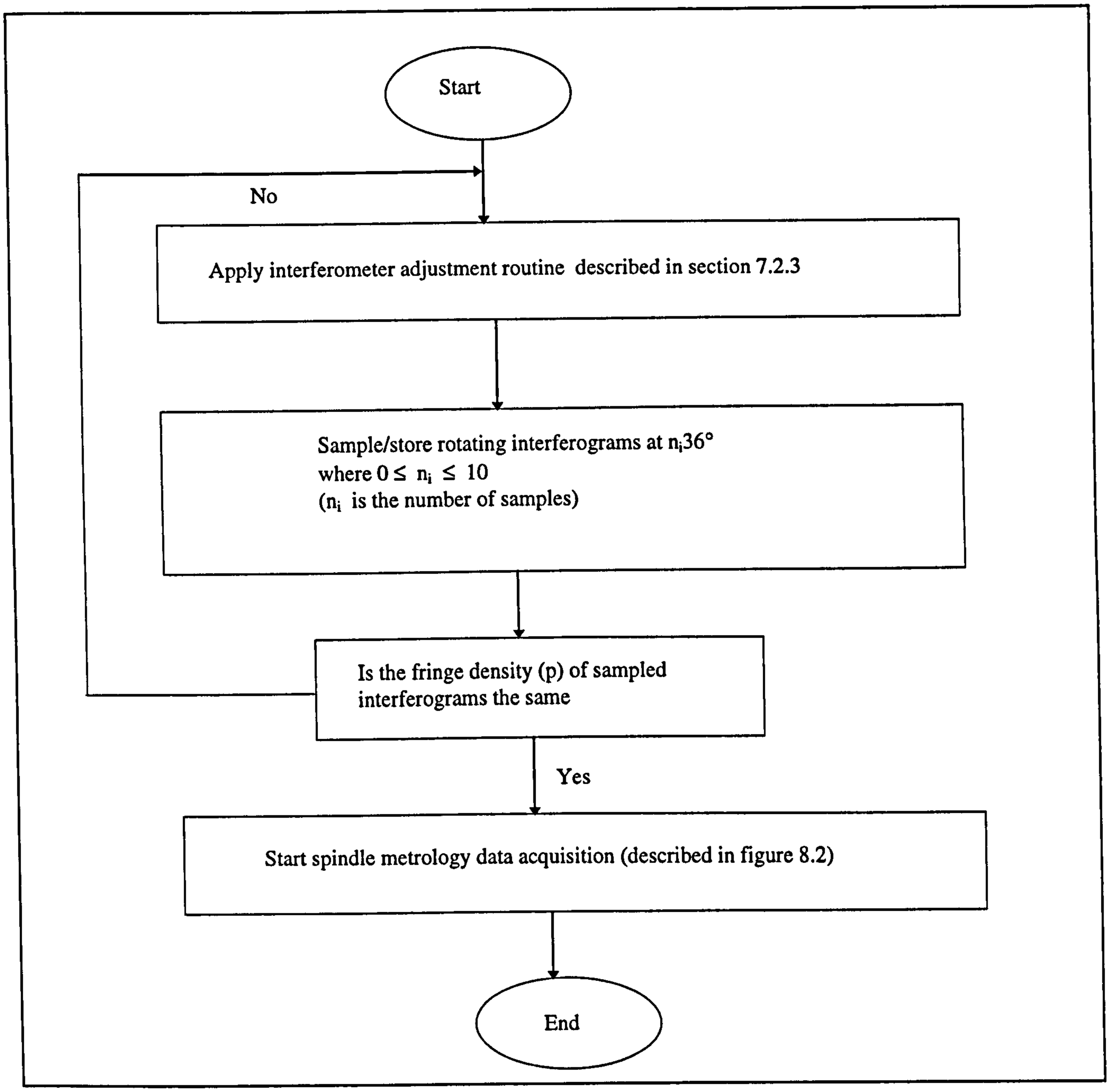


Figure 8.1: Interferometer adjustment routine

8.3.2 MEASUREMENT OF AXIAL ERROR MOTIONS IN X-Y SPACE

The variations in fringe phase of morphologically simplified interferograms (ie see section 6.4.7 for a detailed description of fringe morphology) is a measure of axial error motion in the spatial domain. An algorithm was developed to quantify the difference in phase of the *measured* (with spindle error present) and *reference* (without spindle error) interferograms. This measurement involves scanning both interferograms normal to their orientation.

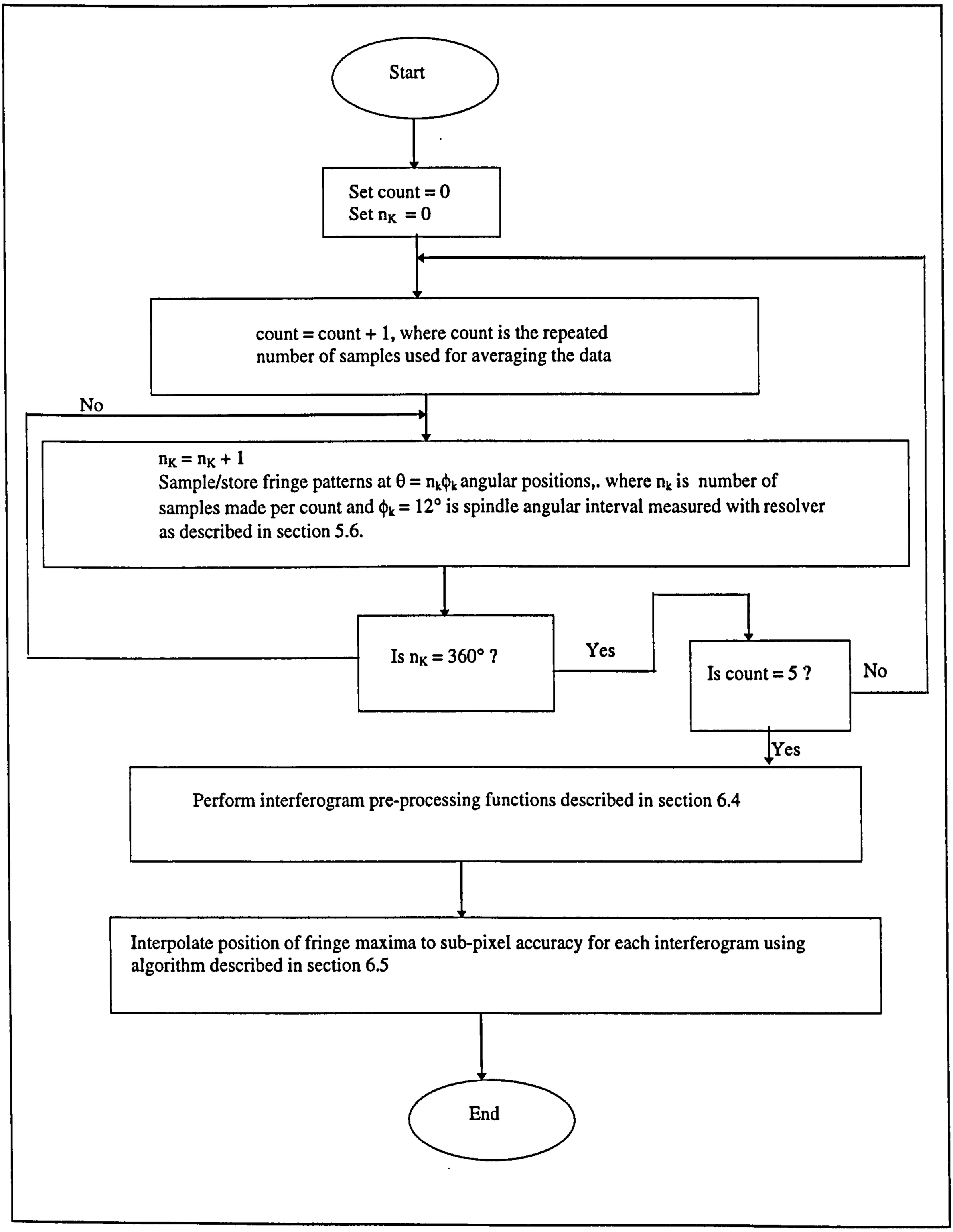


Figure 8.2: Spindle metrology data acquisition routine

The axial error motion (Z_a) is calculated at a chosen spindle position (ϕ) as the ratio of the mean value of all the discrete (N) phase variations along the line scan (Δ_N) and the fringe spacing of the reference interferogram (S_R).

It is defined mathematically as:

$$d_z(\phi,f) = \frac{\overline{\Delta}}{S_R}\lambda \qquad (8.1)$$

where, f: spatial frequency
λ: wavelength of light source
$\overline{\Delta}$: average spatial phase

Figure 8.3 shows the flow chart of the algorithm used for measuring the spindle axial error motion. Measurement of Δ_N in X-Y space is shown schematically in figure 8.4, the measured fringe pattern (shown as dotted lines) is shown superposed on the reference interferogram (shown as solid lines), and the difference between these lines is:

$$\Delta_N = X_{RMAX}(\phi', N) - X_{MMAX}(\phi, N) \qquad (8.2)$$

where, $X_{RMAX}(\phi', N)$ and $X_{MMAX}(\phi, N)$ are the interpolated *reference* and *measured* values of the fringe maximum position of the N^{th} fringe (see section 6.5 for details on the fringe interpolation routine). ϕ' and ϕ are the angular positions for the *reference* and *measured* interferograms respectively.

The mean value of fringe phase $\overline{\Delta}(\phi)$ is:

$$\frac{1}{N}\sum_{N=1}^{N}\Delta_N \qquad (8.3)$$

The influence of spindle axial error motion on the intensity profile of the measured fringe pattern can be defined mathematically as:

$$I(X_{MMAX}, \phi, N) = 2I_0(1+\cos[kX(\phi', N) + \overline{\Delta}(\phi)]) \qquad (8.4)$$

where $k = \frac{2\pi}{\lambda}$: wave propagation number

This is also described schematically in figure 8.5a, where the superposition of the reference and measured interferograms indicate a phase shift due to the presence of spindle axial error motion.

A number of issues were considered regarding the measurement of spindle axial error motion in X-Y space. These include:

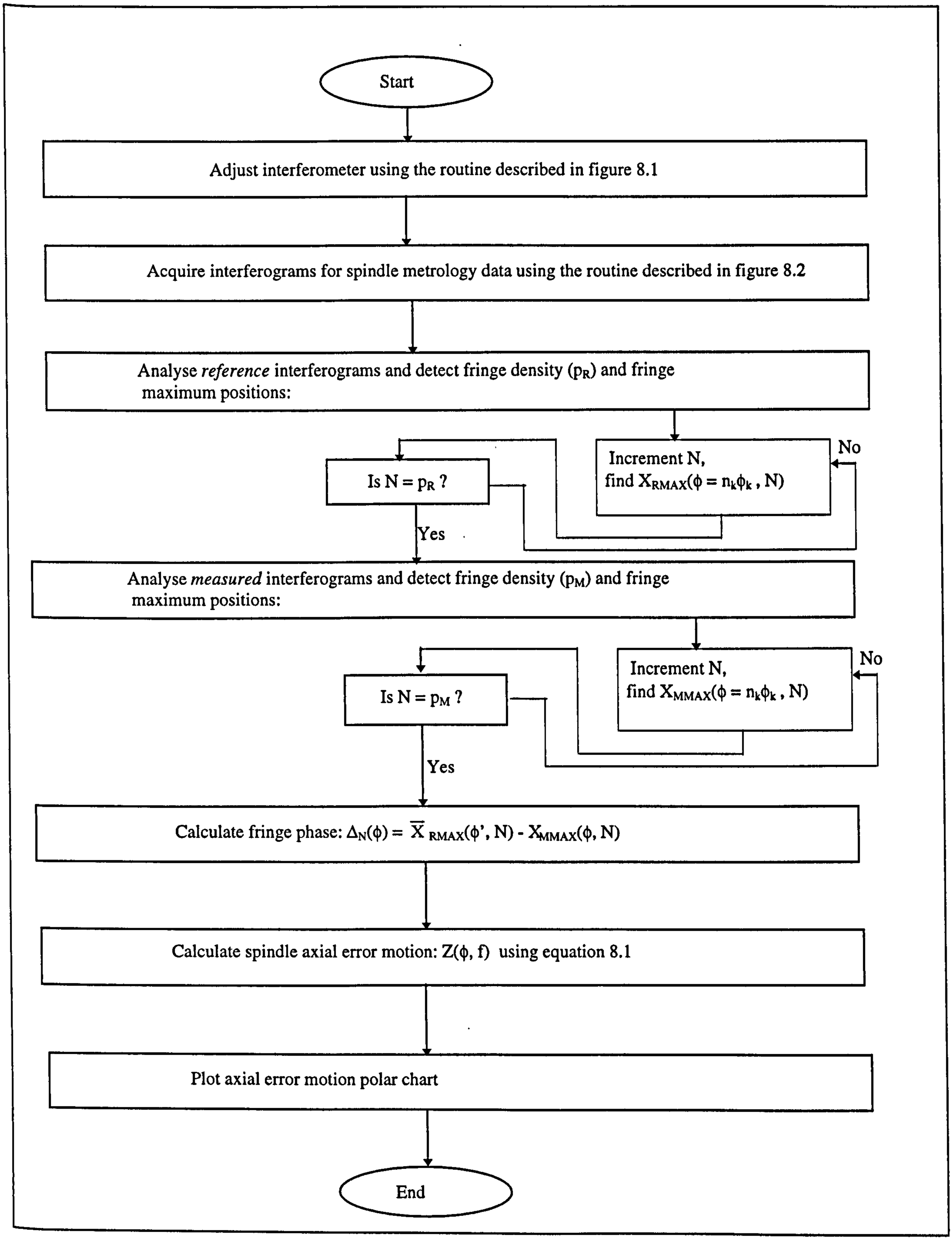


Figure 8.3: Routine for measuring spindle axial error motion

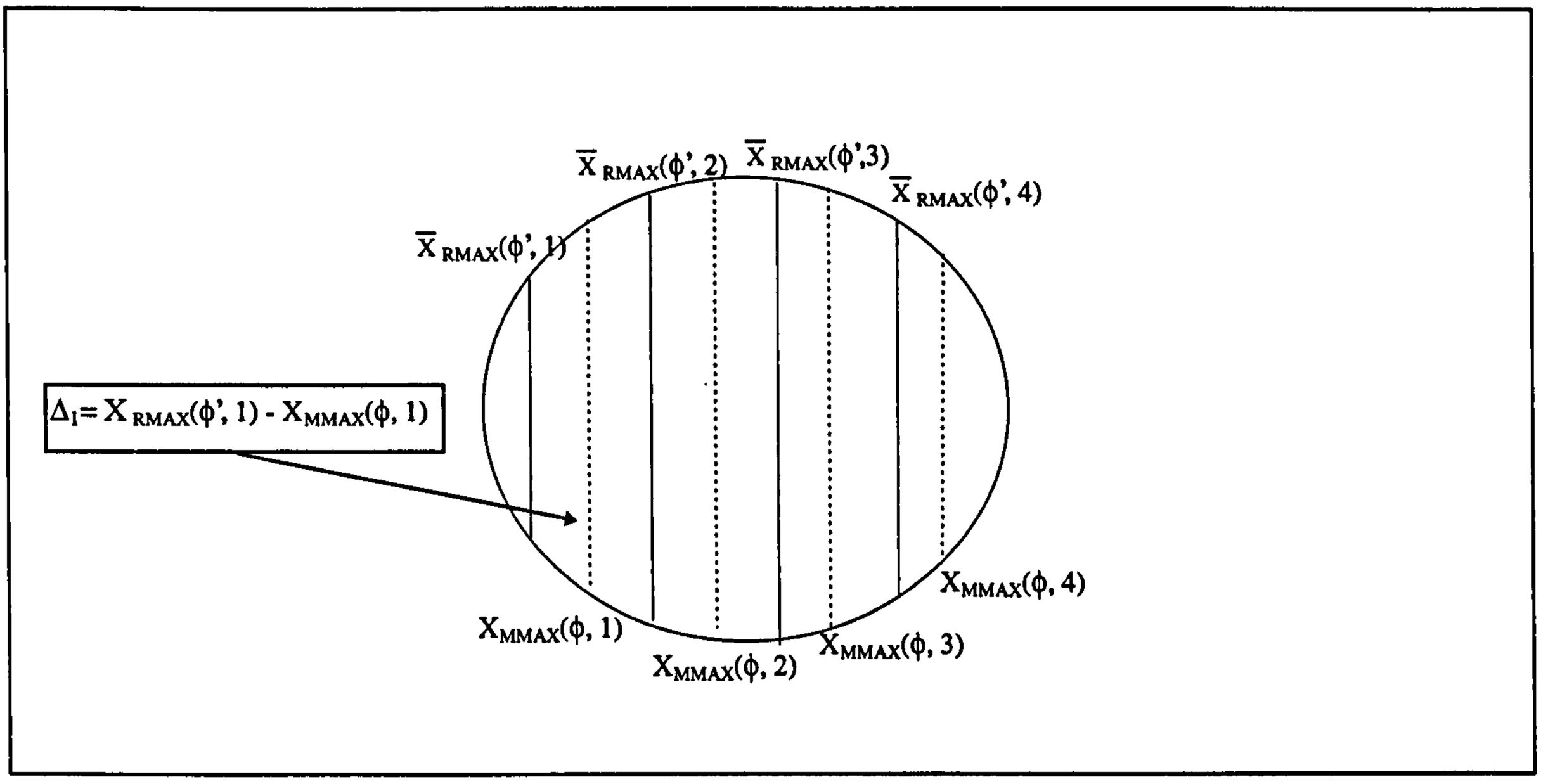


Figure 8.4: Measurement of fringe phase

- limitation in fringe spatial resolution which reduces the number of pixels between two fringe peaks (fringe spacing)

- error introduced when the interferogram is rotated (ie rotation of the interferogram is required in order to scan it at orientations greater than 0 degrees). This error is due to the non-unity aspect ratio of the area occupied by the fringe pattern and it will affect the detection of the fringe peak position

- influence of spindle tilt errors on the measurement of axial error motion

In this work, the average fringe density (number of fringes per interferogram frame) used for spindle metrology was 8. Each interferogram had a limited spatial resolution of 58×90 pixels (ie an aspect ratio of 0.65) as shown in figure 8.6a. This was a problem, because it gave a limited measurement resolution (ie a fringe spacing of 7 pixels), equivalent to axial error measurement resolution of 90 nm. In order to resolve this problem an area of size 30×30 pixels was extracted from the interferogram and re-mapped to spatial resolution of 256×256 as shown in figure 8.6b. This modified the fringe nominal period from 7 to 54 pixels, hence providing a measurement resolution of 11 nm. The only draw-back with this approach is the average fringe spacing data (as required in equation 8.3), was calculated from half the fringe density of the original (low resolution) interferogram.

The inaccuracy introduced in the axial error measurement due to the non-unity aspect ratio of the interferogram was reduced by analysing a square area of 30×30 pixels (ie with an aspect ratio of 1), extracted from the original area of interest of size 58×90 pixels (ie with aspect ratio of 0.65).

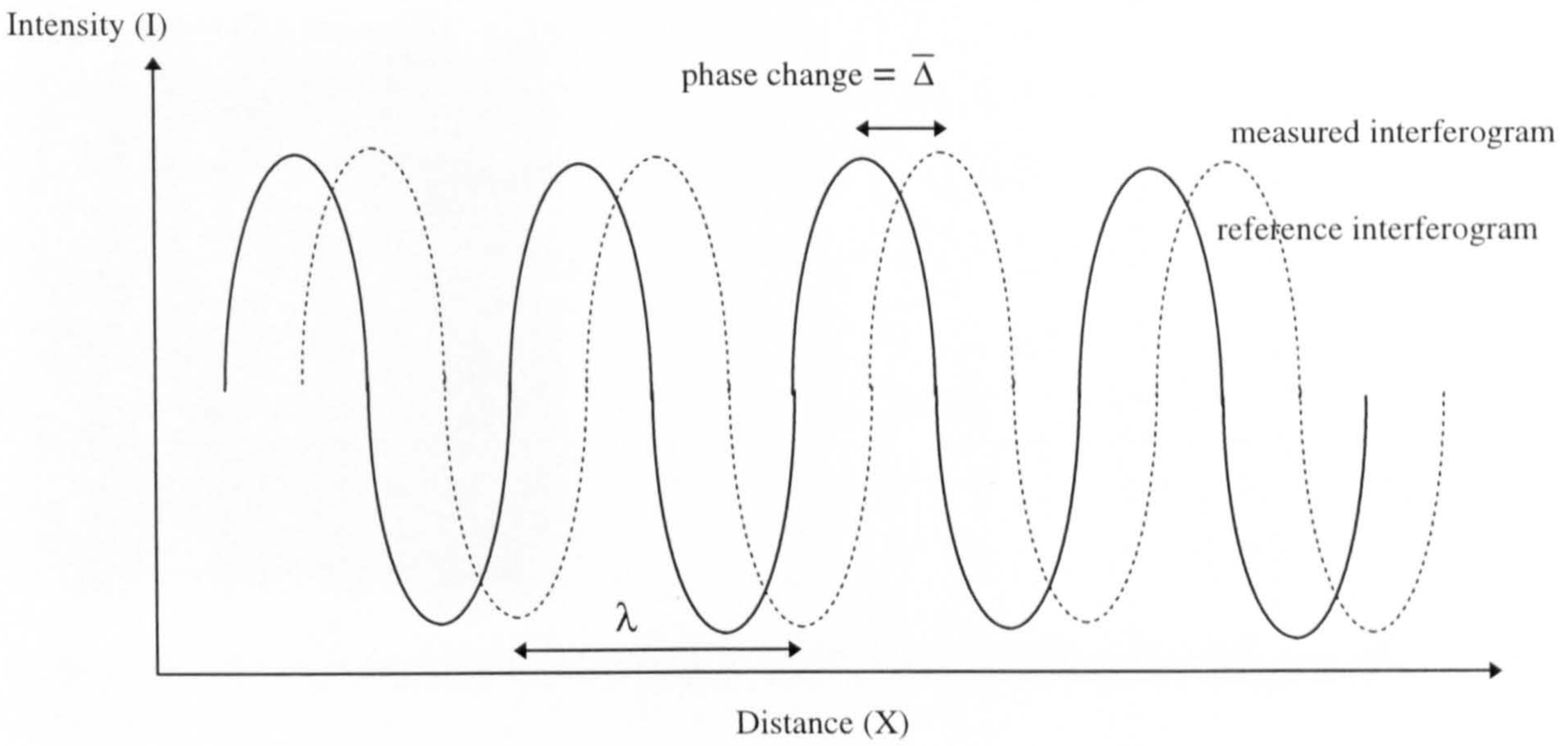


Figure 8.5a: Measurement of fringe phase change in the spatial domain

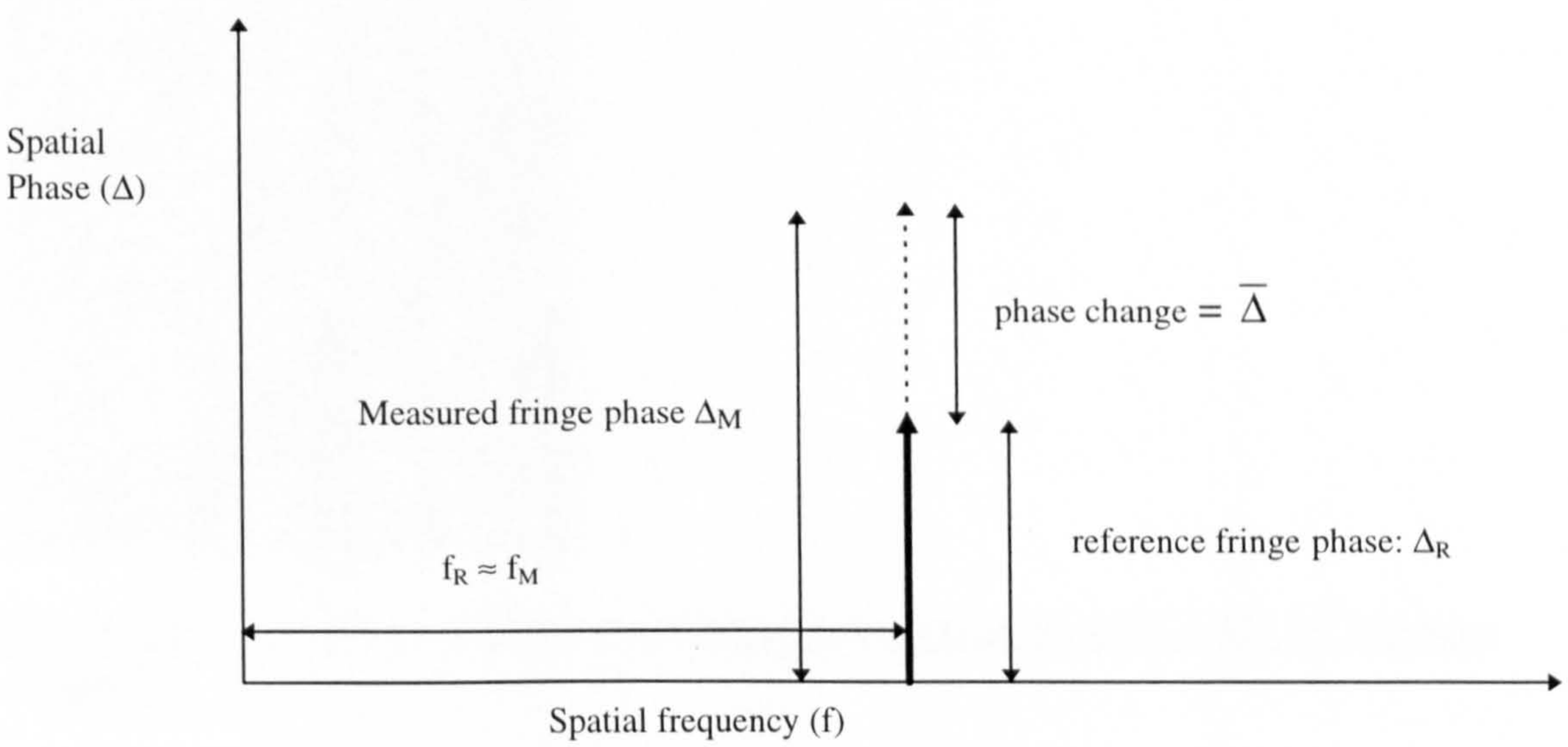


Figure 8.5b: Measurement of fringe phase change in the Fourier domain

Figure 8.5: Measurement of fringe phase change in the spatial and Fourier domain

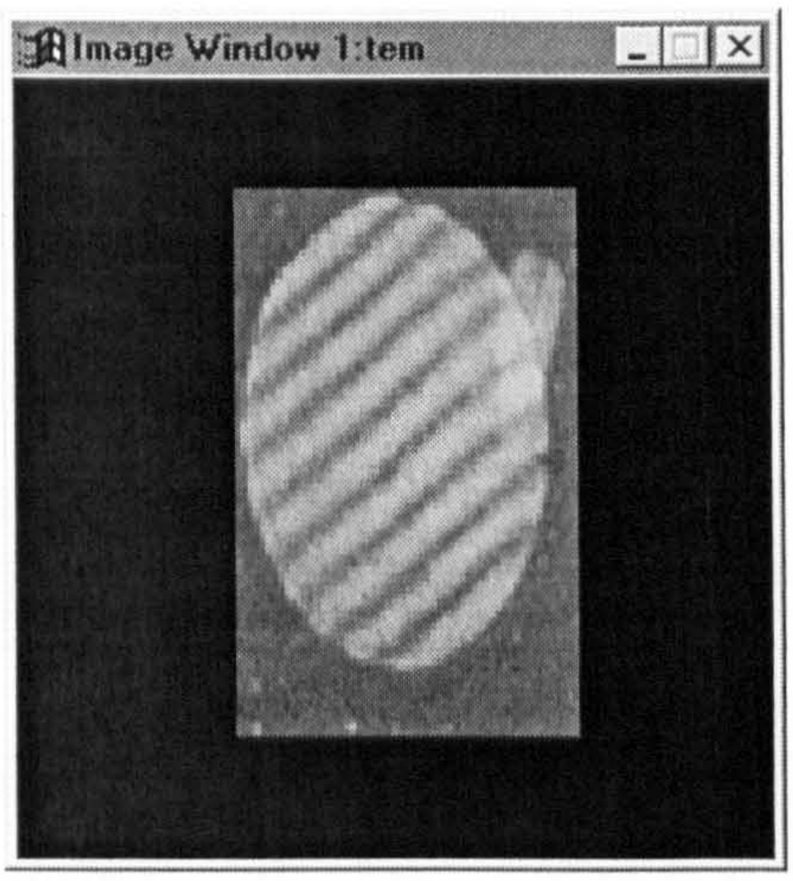


Figure 8.6a: Limited spatial resolution of interferogram with a 58 × 90 *area of interest*

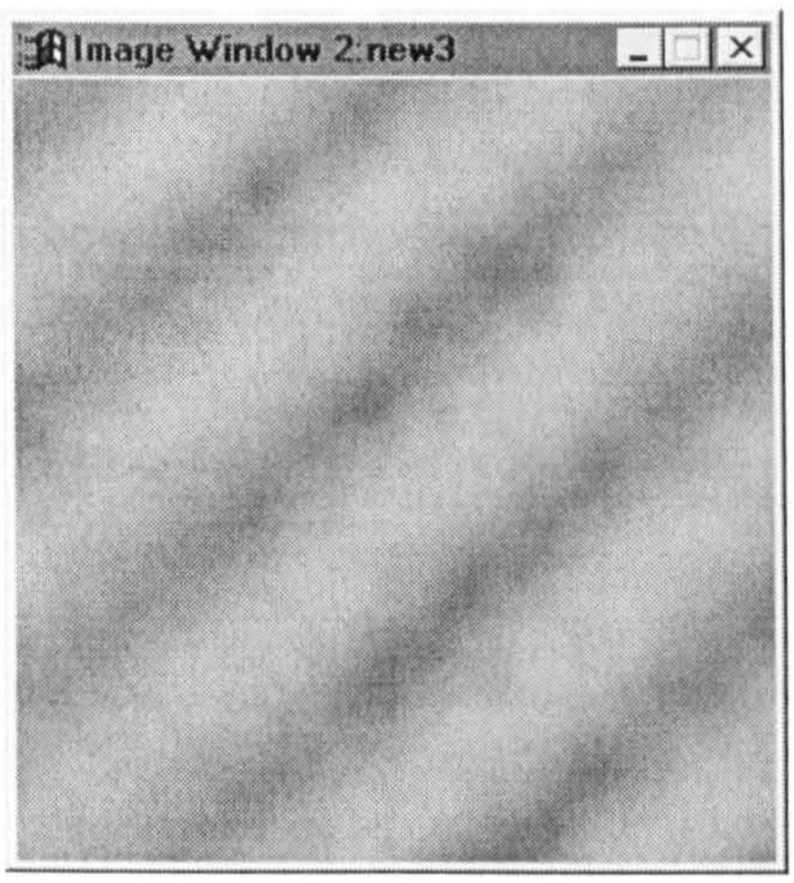


Figure 8.6b: Improved spatial resolution of interferogram with a 256 by 256 *area of interest*

In order to suppress the influence of angular errors (X and Y tilts) on the measurement of axial error motion, the difference in fringe spacing, (S_c) between the reference (S_R) and measured (S_M) interferogram was subtracted from the measured fringe phase.

The axial error motion obtained by modifying equation 8.1 after applying the above corrections is:

$$Z_c = \frac{\left[\frac{1}{n}\sum_{n=1}^{n}\left(X_{CRMAX} - X_{CMMAX}\right) - \left(S_R - S_M\right)\right]\lambda}{S_R} \qquad (8.5)$$

$$= \frac{(\overline{\Delta}_c - S_c)\lambda}{S_R} \qquad (8.6)$$

where, X_{CRMAX}, X_{CMMAX}: modified reference and measured fringe peak positions
S_R, S_M: modified reference and measured fringe spacing
Δ_c: modified fringe phase

8.3.3 MEASUREMENT OF AXIAL ERROR MOTIONS IN FOURIER SPACE

An alternative way of calculating the spindle axial error is to use the phase information obtained from a fast Fourier transform (FFT) of the interferogram. The main benefit of this approach is that the accuracy of the fringe phase calculation is not limited by the fringe pattern spatial resolution.

The influence of spindle axial error on the spatial phase of the measured and reference interferogram is shown schematically in figure 8.5b, where the dotted and solid lines represent the phase values in Fourier space of the reference and measured interferograms respectively.

The phase component (Δ) of an interferogram in Fourier (U,V) space can be defined mathematically as:

$$\Delta(U_s^-, V_s^-) = \Delta(U_s^+, V_s^+) = \tan^{-1}\left(\frac{I^r(U_s^+, V_s^+)}{I^i(U_s^+, V_s^+)}\right) \qquad (8.7)$$

where, I^r, I^i : real and imaginary components of FFT interferogram
U_s^-, V_s^- : positions of the negative first harmonic
U_s^+, V_s^+ : positions of the positive first harmonic

The measurement of the fringe phase in 2-dimensional Fourier space requires the co-ordinates of interferogram first harmonics. This is calculated from an a priori knowledge of the interferogram spatial frequency, this was also obtained in Fourier space (see section 6.4.6 for a detailed description). Figure 8.7 shows a plot of fringe pattern phase against spatial frequency for one of the acquired interferograms after a 2-dimensional FFT is performed.

Calculation of the fringe pattern phase at its fundamental frequency (first harmonic) is usually affected by errors due to the quasi-sinusoidal nature of the interferogram intensity pattern, this explains the random but periodic nature of the interferogram phase shown in figure 8.9. In this work the fringe phase evaluation is not based on this quasi-sinusoidal interferogram (grey-level) function but rather on its bi-level transform (ie obtained by binarisation as described in section 6.4.5) which is a square function (ie a summation of sinusoids of different Fourier harmonics). By fitting a sinusoidal function to the one-dimensional intensity pattern of the bi-level interferogram it is possible to accurately calculate the fringe intensity (amplitude) and phase at its fundamental frequency.

The quasi-sinusoidal function of the interferogram can be defined as:

$$I_q = 2I_0(1 + \cos\Delta) = 2I_0(1 + \cos(fx)) = a_0 + b\cos(fx) \qquad (8.8)$$

where a_0: interferogram dc offset
b: interferogram amplitude
f: spatial frequency
x: interferogram one-dimensional space
Δ: phase difference

The bi-level transform of equation 8.8 is a square function defined as:

$$I_b = a_0 + \sum_{k=1}^{\infty}[a_k\cos(kfx) + b_k\sin(kfx)] \qquad (8.9)$$

where a_k, b_k: interferogram amplitudes at different harmonics
k: harmonic number

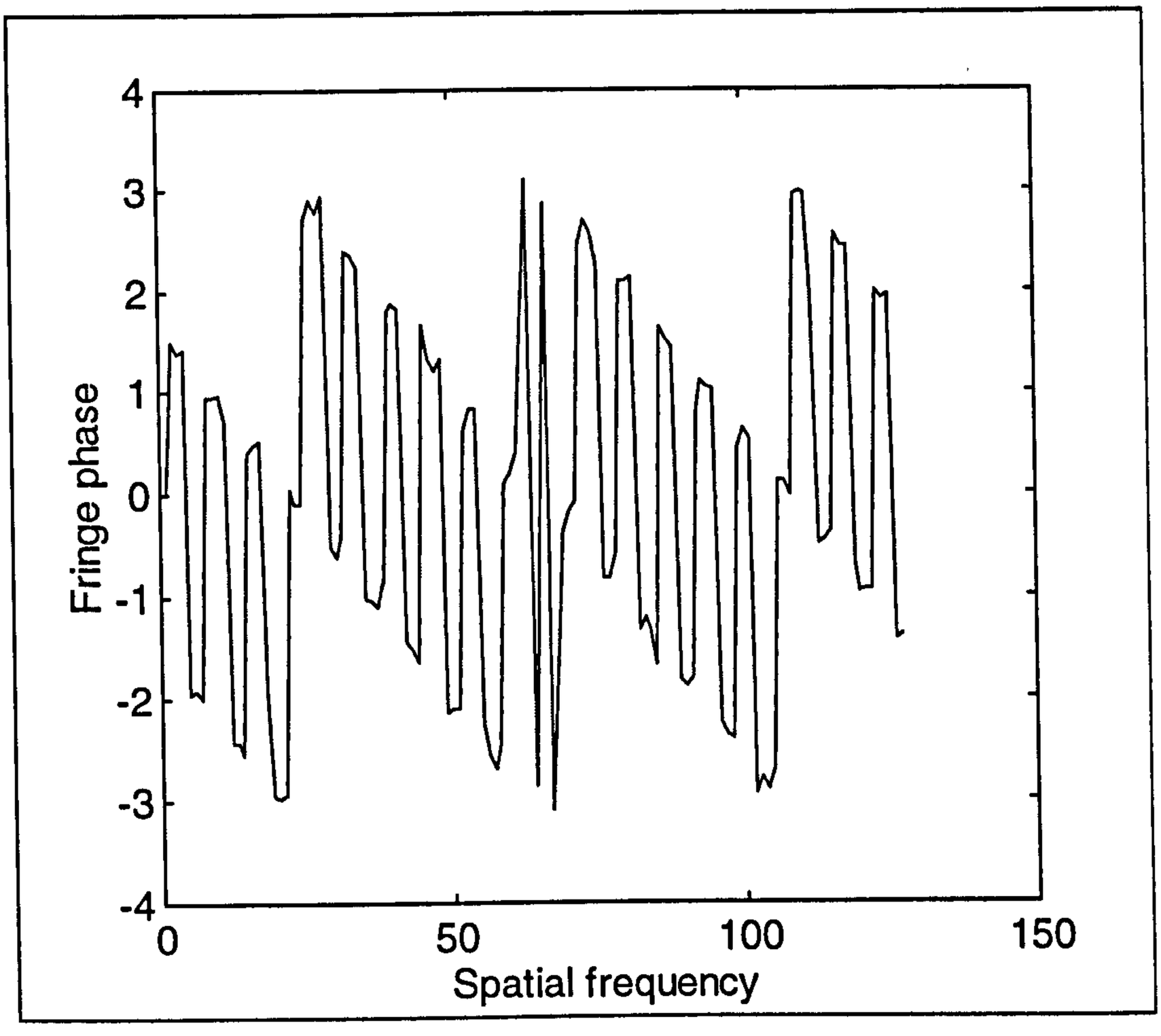


Figure 8.7: Interferogram phase versus spatial frequency

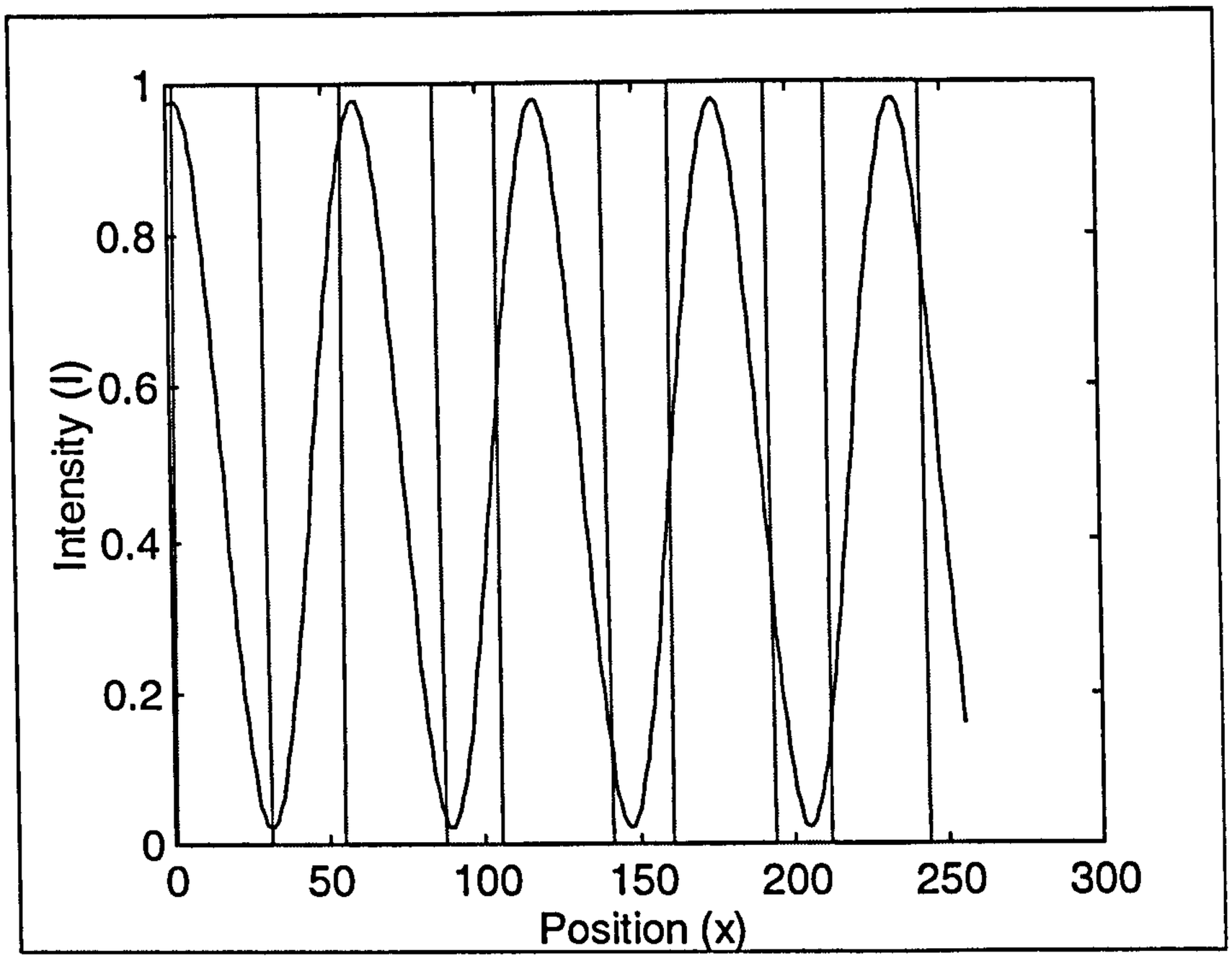


Figure 8.8: Fundamental frequency sinusoidal fit to a bi-level interferogram

At the fundamental frequency $f = \frac{2\pi}{S}$ and at the first harmonic, k = 1, equation 8.8 gives a perfect sinusoidal fit to the square function:

$$I_s = a_0 + a_1\cos(fx) + b_1\sin(fx) = a_0 + \sqrt{a_1^2 + b_1^2}\cos\left(\frac{2\pi x}{S} + \Delta\right) \qquad (8.10)$$

which has a fringe phase: $\Delta = \tan^{-1}\left(\frac{b_1}{a_1}\right)$ (8.11)

Figure 8.10 indicates the (fundamental frequency) sinusoidal (shown in blue) fit for a binarised interferogram with its bi-level intensity profile (shown in red).

An algorithm based on equations 8.10 and 8.11 was developed to evaluate the phase of the reference and measured interferograms. The phase difference between the measured and reference interferograms is:

$$\Delta_R(\phi') - \Delta_M(\phi) - \Delta_C(\phi) = \overline{\Delta}(\phi) \qquad (8.12)$$

where, Δ_M, Δ_R: are the phase values of the measured and reference interferograms

Δ_C: phase correction term

and this was used to calculate the spindle axial error as:

$$\left(\frac{\overline{\Delta}(\phi)}{2\pi}\lambda\right) \qquad (8.13)$$

8.4 MEASUREMENT OF TILT ERROR MOTIONS

8.4.1 INTRODUCTION

The tilt error motion vector (Γ_{xy}) can be calculated in the spatial or Fourier domain. The following sub-sections describe this.

8.4.2 MEASUREMENT OF TILT ERROR MOTIONS IN X-Y SPACE

The variation of fringe spacing (S) in time of morphologically simplified interferograms is a measure of tilt error motion in the spatial domain. An algorithm was developed to detect the progressive change in fringe spacing of the measured interferogram with respect to the reference interferogram. Figure 8.9 shows the flow chart of the algorithm. Tilt error motion vector is calculated as the ratio of the mean

difference in fringe spacing (δ_N) between the measured and reference interferograms, and the mean fringe spacing ($\overline{S}^R$) of the reference interferogram. The tilt error motion vector is defined mathematically as:

$$\Gamma_{xy} = \frac{\delta_N}{\overline{S}^R}\lambda \qquad (8.14)$$

The difference in fringe spacing between the measured and reference interferograms is shown in figures 8.10a and 8.10b. Figure 8.10b shows an expanded view of figure 8.13a indicating reference fringe spacing S_N^R (shown as a solid line) and measured fringe spacing S_N^M (shown as a dotted line). There is also an angle ξ formed between S_N^R and S_N^M, due to the occurrence of the spindle tilt in a direction non- parallel to the orientation of the reference interferograms (as shown in figure 8.10b). ξ is measured as the difference in orientation between the reference and measured interferograms:

$$\xi = \phi_R - \phi_M \qquad (8.15)$$

θ_R and θ_M are the orientations of the reference and measured interferograms measured using the FFT method described in section 6.4.4.6.

The difference between S_N^R and S_N^M in the direction parallel to the orientation of the reference interferogram θ_M is:

$$\delta_N = \left(\frac{S_N^R}{\cos\xi} - S_N^M\right) \qquad (8.16)$$

and its mean value $\overline{\delta}_N$ is:

$$\frac{1}{N}\sum_{N=1}^{N}\delta_N \qquad (8.17)$$

The fringe spacing of the reference interferogram as shown in figure 8.10b is computed as:

$$S_N^R = X_{RMAX}(\phi, N) - X_{RMAX}(\phi', N+1) \qquad (8.18)$$

Similarly the fringe spacing of the measured interferogram is:

$$S_N^M = X_{MMAX}(\phi, N) - X_{MMAX}(\phi', N+1)$$
(8.19)

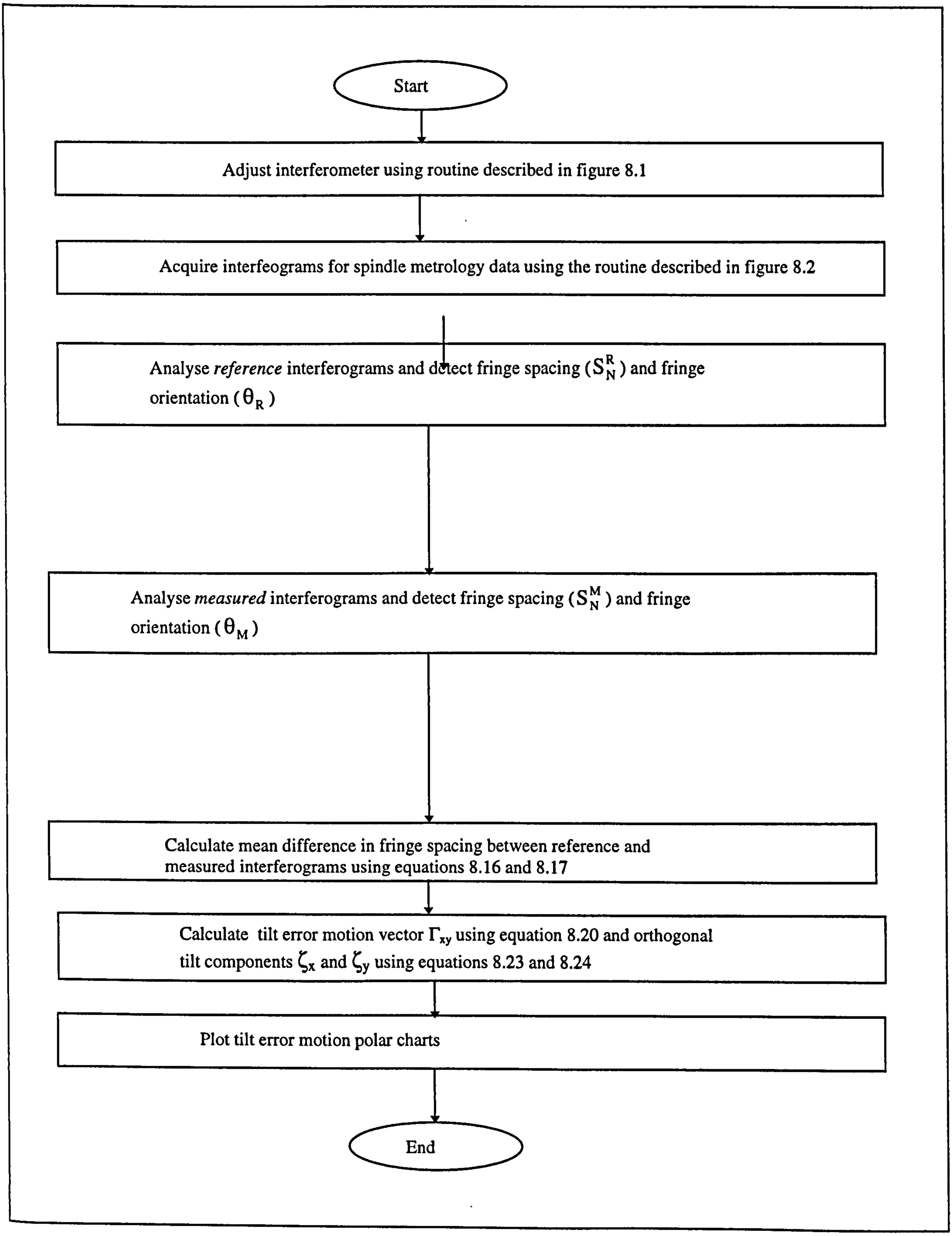


Figure 8.9: Measurement spindle tilt error motions in the spatial domain

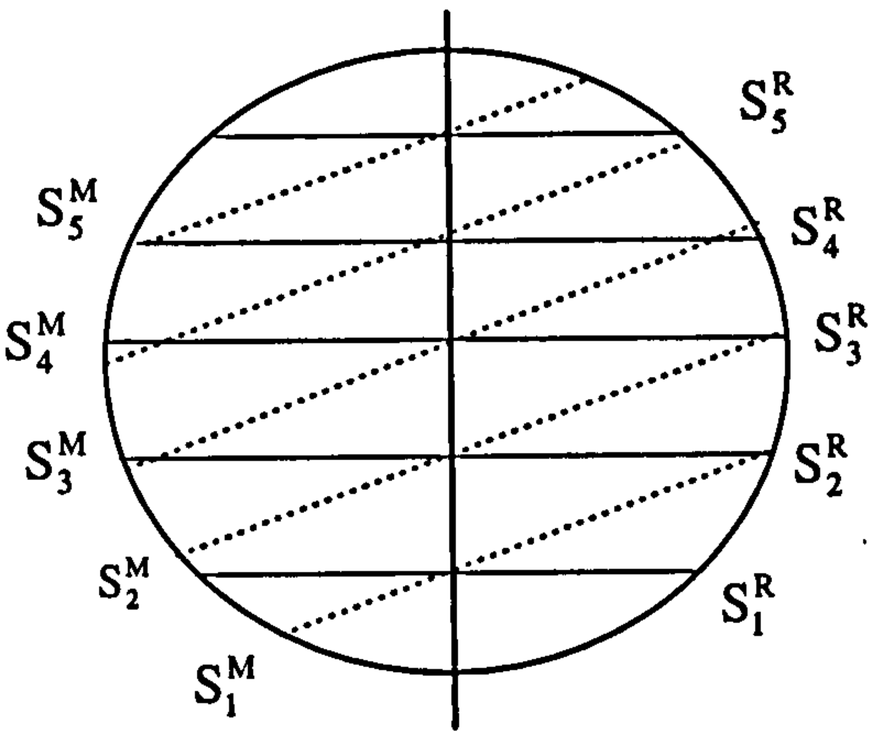


Figure 8.10a : Superposition of reference and measured interferograms

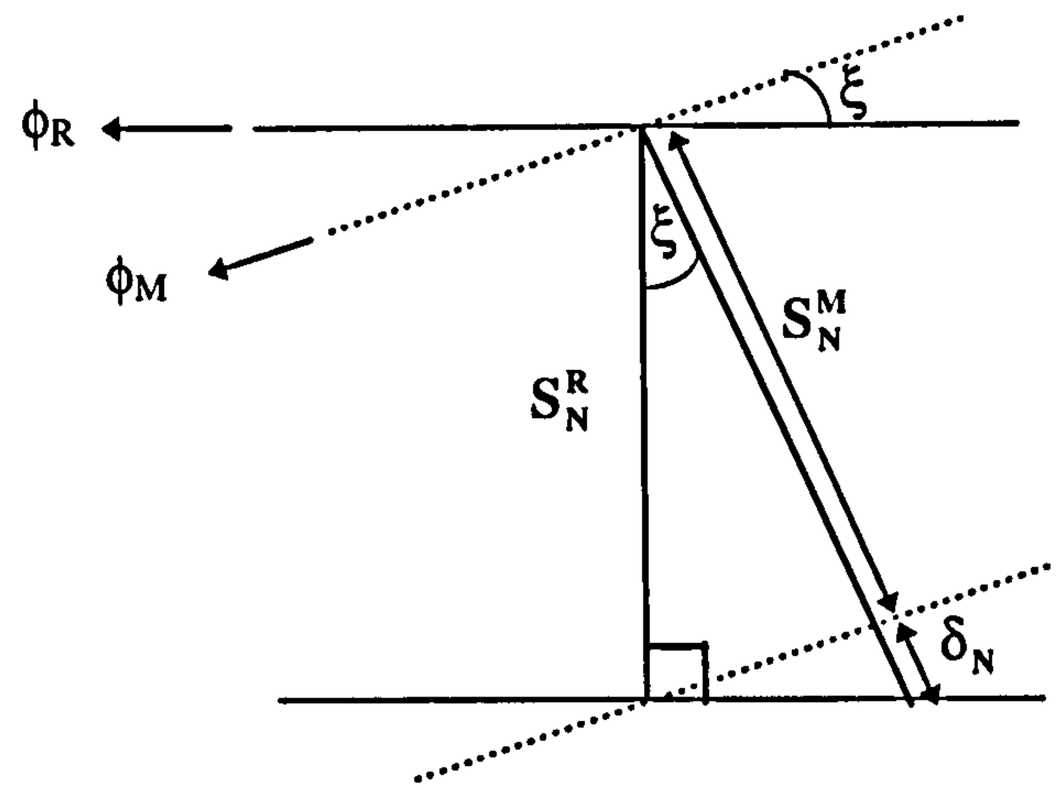


Figure 8.10b: Difference in fringe spacing of reference and measured interferograms

Figures 8.10a and 8.10b: Measurement of fringe spacing difference

8.4.3 MEASUREMENT OF TILT ERROR MOTIONS IN FOURIER SPACE

An alternative way of calculating Γ_{xy} is by using the spatial frequency measurements of the reference and measured interferograms as shown in figure 8.11a. The spatial frequency (f) of the sampled interferograms is the averaged valued of the reciprocal of the fringe period (spatial frequency) as shown in figure 8.11b. Hence equation 8.16 can be modified to:

$$\Gamma_{xy} = \frac{\aleph}{f_R}\lambda \qquad (8.20)$$

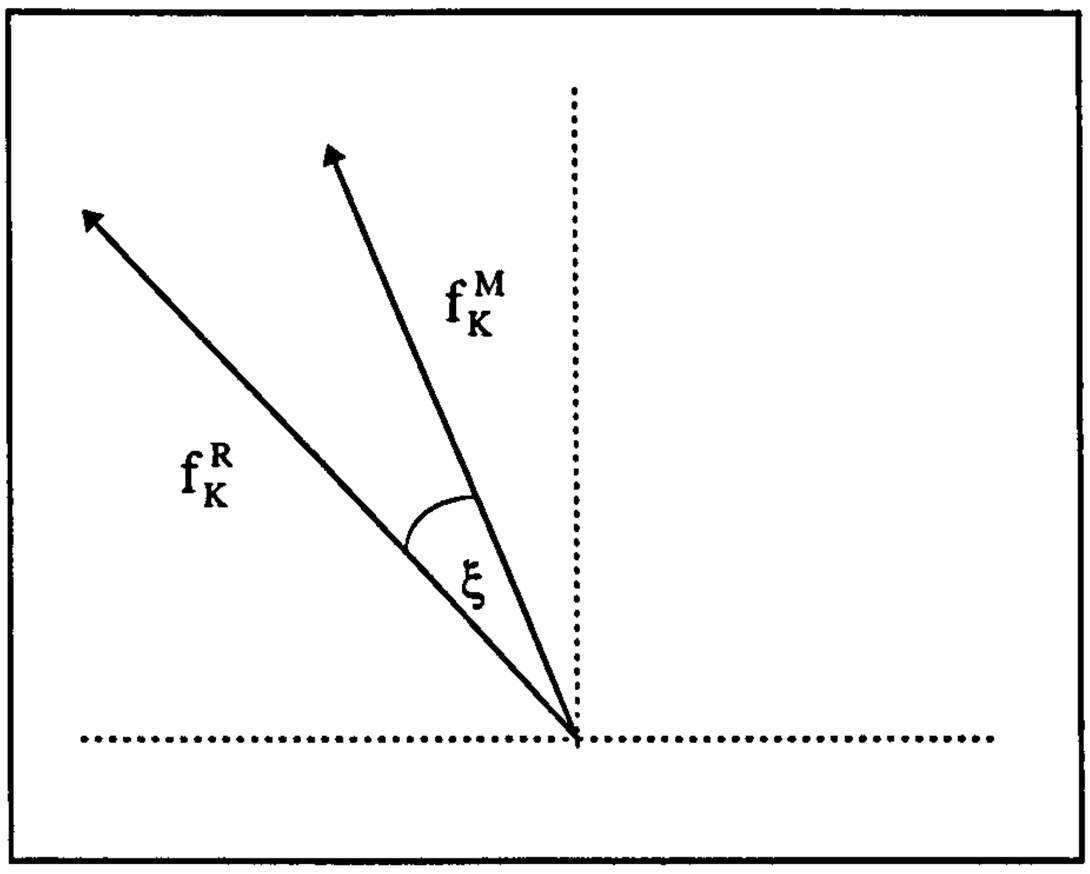


Figure 8.11a: Spatial frequency vectors of measured and reference interferograms and the angle between them.

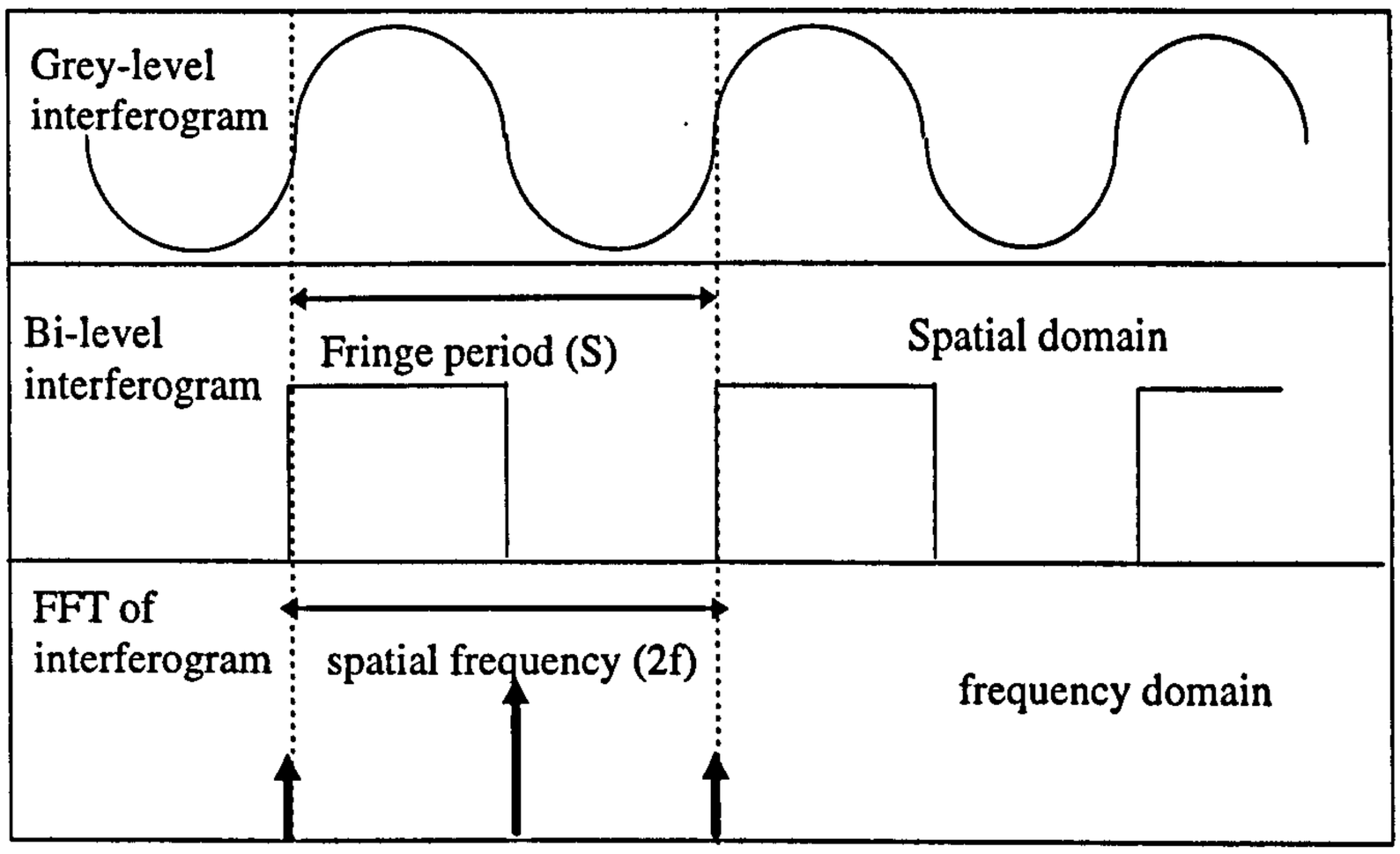


Figure 8.11b: Relationship between fringe spatial frequency and period

Figure 8.11a and 8.11b: Description of fringe spatial frequency

$$\aleph = \left(\frac{f_R}{\cos \xi} - f_M \right) \tag{8.21}$$

where, $\aleph$: difference in spatial frequency between the reference and measured interferogram
f_R: spatial frequency of the reference interferogram
f_M: spatial frequency of the measured interferogram

The influence of spindle tilt error motion on the fringe spacing of the measured fringe pattern can be defined mathematically as:

$$I(X_{MMAX}, \phi, N) = 2I_0\{1+\cos[f_R(\phi', N) + \aleph(\phi')]X\} \tag{8.22}$$

The tilt error vector Γ_{xy} is resolved into two orthogonal contributions:

$$x_\zeta = \Gamma_{xy} \cos \theta_K^M \tag{8.23}$$
$$y_\zeta = \Gamma_{xy} \sin \theta_K^M \tag{8.24}$$

about the X and Y axes of the spindle as shown in figures 8.12a - 8.12c. Orthogonal tilt error motions are measured as angular movements defined as:

$$E_x = \tan^{-1}\left(\frac{x_\xi}{D}\right) \approx \left(\frac{x_\xi}{D}\right) \tag{8.25}$$

$$E_y = \tan^{-1}\left(\frac{y_\xi}{D}\right) \approx \left(\frac{y_\xi}{D}\right) \tag{8.26}$$

respectively, where D = 100 mm: is the diameter of the spindle face-plate.

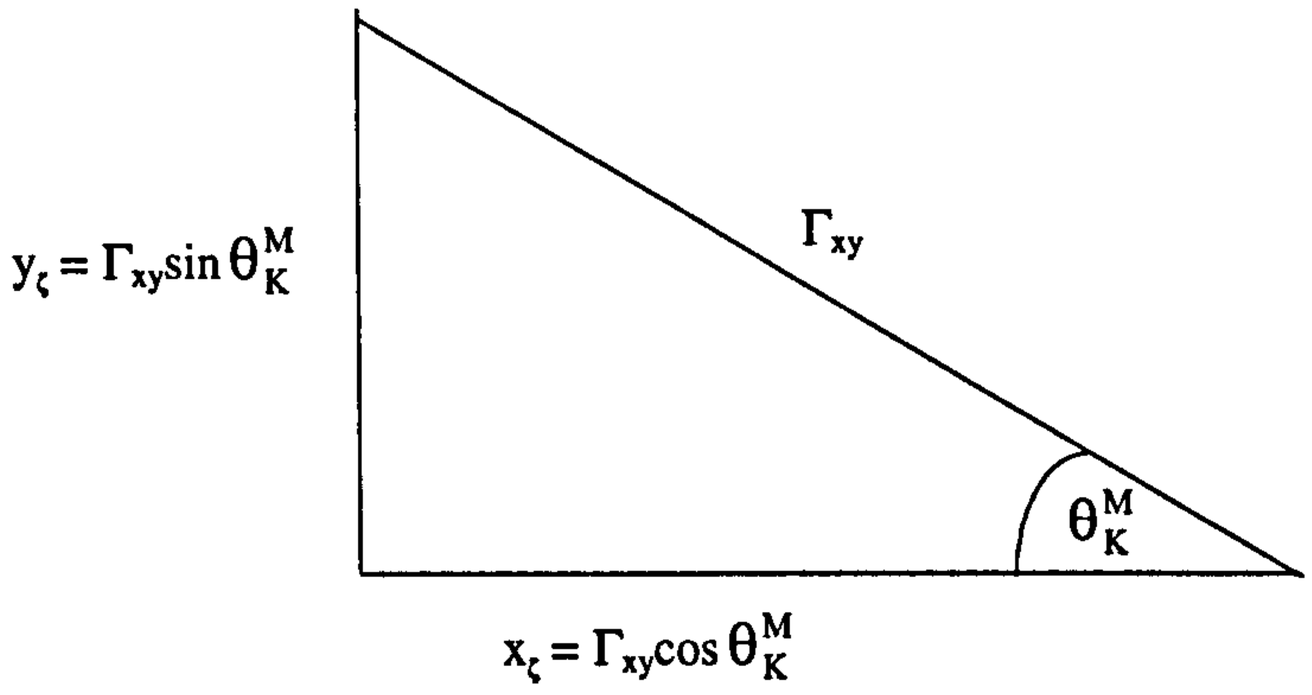


Figure 8.12a: Tilt vector and its resolved components

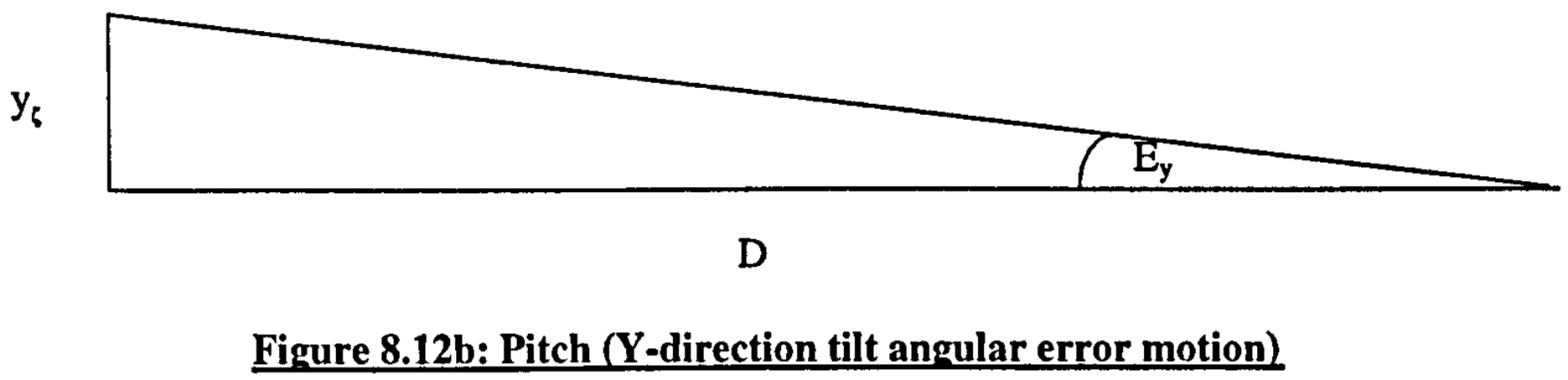


Figure 8.12b: Pitch (Y-direction tilt angular error motion)

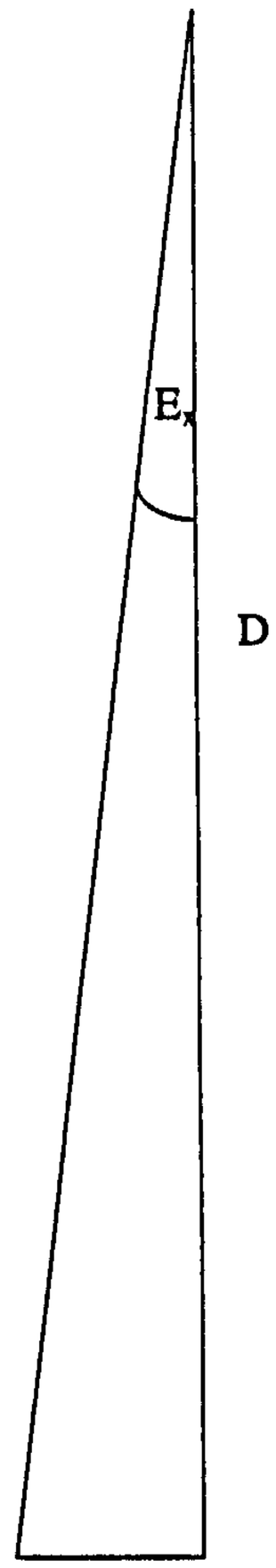


Figure 8.12c: Roll (X-direction tilt angular error motion)

Figures 8.12a, 8.12b and 8.12c: Calculation of the orthogonal tilts

8.5 PRESENTATION OF ERROR MOTIONS

8.5.1 INTRODUCTION

According to ANSI/ASME American Standard B89.3.4M (1985), a useful form for displaying error motion measurements is by using a polar chart. In this section, description of the error motion polar chart and type of error motion plots used in spindle metrology are given. A brief description of error-motion noise and its reduction using digital filtering is given. Finally a detailed description of the results obtained in this work is given.

8.5.2 ERROR MOTION POLAR CHART

An error motion polar chart is a plot of error motion vector (ι) against the angular position (θ) of the spindle. Generally the error motion value obtained from a polar chart is equal to the difference in radii between two concentric circles that will just enclose the corresponding error motion polar plot, and the value obtained depends on the location of the common centre of the two circles [ANSI/ASME American Standard B89.3.4M (1985)]. It is important to define the centre of theses two circles, they are usually referred to as *polar centres*.

There are 5 types of polar chart centres, theses are:

- least square circle (LSC)
- maximum inscribed circle (MIC)
- minimum radial separation (MRS)
- minimum circumscribed circle (MCC)
- polar centre (PC)

In this work the *polar plot* centre was used to calculate spindle error motion (e_p) value as the radial difference (R - r) between two concentric circles (peak to valley) whose centres coincide with that of the polar chart (as shown schematically in figure 8.15).

The co-ordinates $X(\theta)$ and $Y(\theta)$ of points on the polar plot are obtained at spindle angular positions (θ), and are defined as:

$$X(\theta) = e_0 + e(\theta)\cos\theta \quad (8.27)$$

$$Y(\theta) = e_0 + e(\theta)\sin\theta \quad (8.28)$$

where, $e(\theta)$: measured axial or tilt error motions at θ

e_0: d.c offset, which is the radius of the *base circle* of the polar chart

The rectilinear co-ordinates of the polar chart are: $R(\theta) = e_0 + e(\theta)$ and θ respectively.

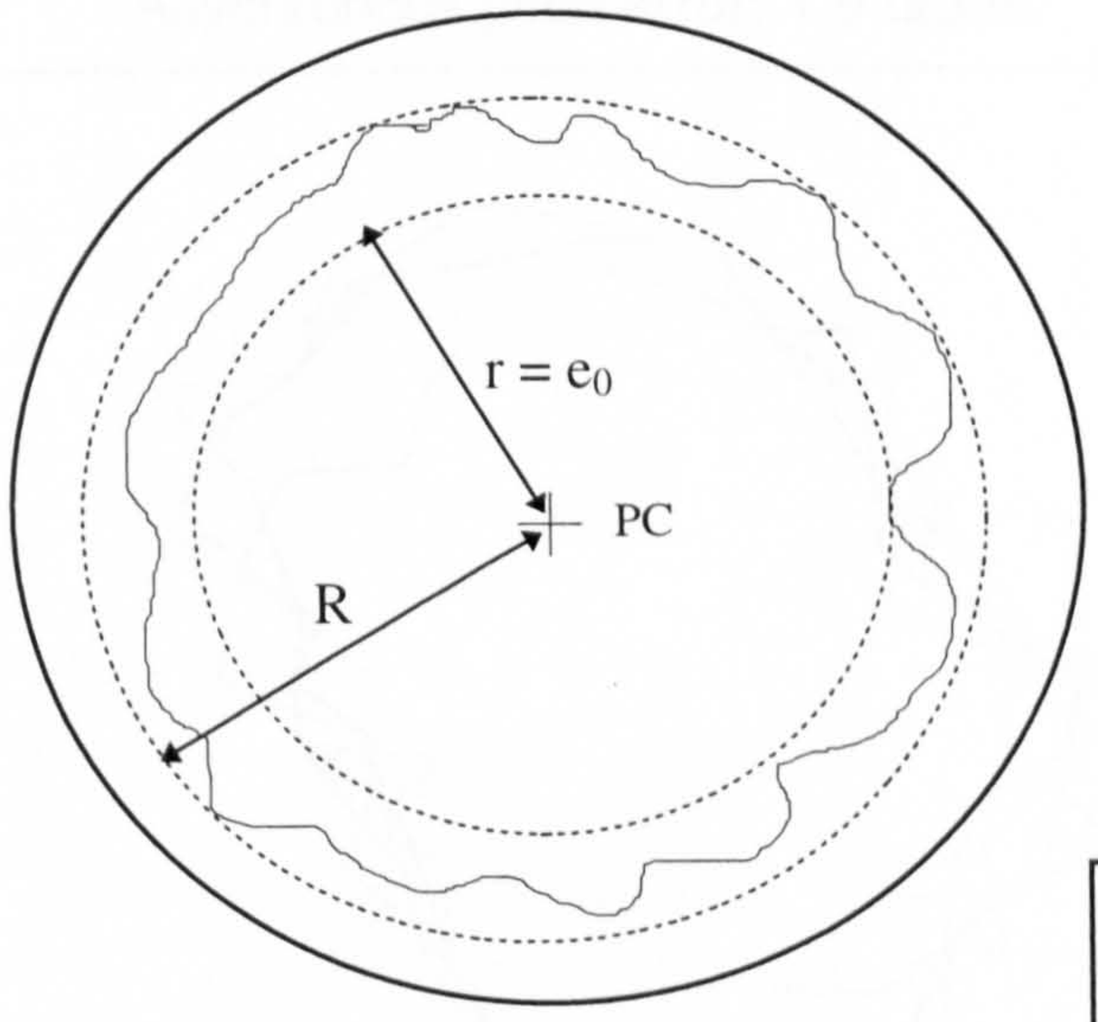


Figure 8.13: Schematic of error motion measurement based on the *polar centre*

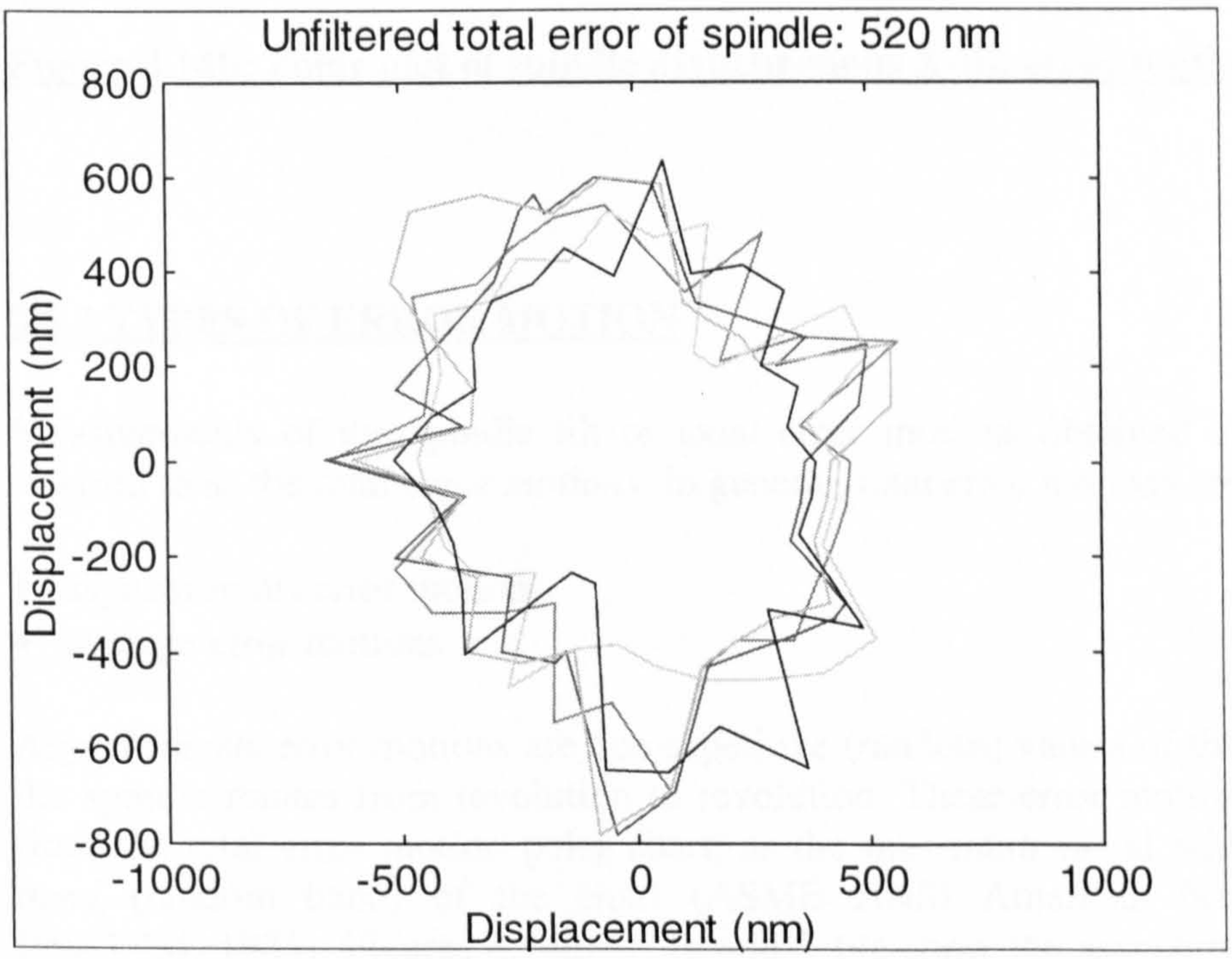


Figure 8.14a: Polar plot of spindle asynchronous axial error motion (for 5 data samples)

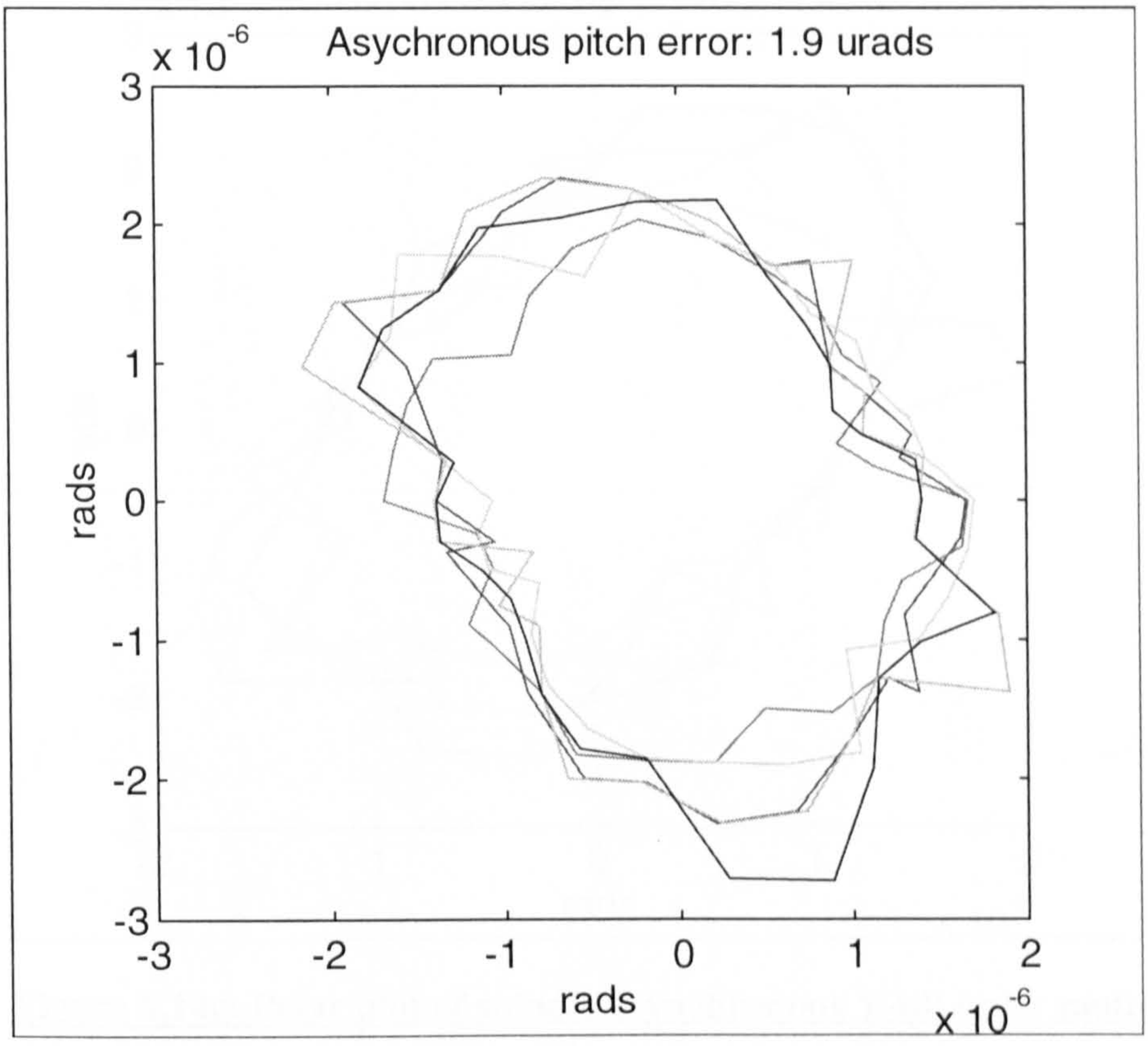


Figure 8.14b: Polar plot of spindle asynchronous X-tilt error motion

8.5.3 TYPES OF ERROR MOTION

Measurements of the spindle tilt or axial error motions obtained in this work are referred to as the *total error motions*. In general, total error motions are categorised as:

- asynchronous error motions
- average error motions

Asynchronous error motions are non-repetitive (random) values of the error vector as the spindle rotates from revolution to revolution. These error motions are measured from the total error motion polar chart, as the maximum radial width of the *cloud band* (random band) of the chart (ASME ANSI American National Standard B89.3.4M, 1985). Figures 8.14a, 8.14b and 8.14c show the asynchronous error plots of the interferometrically measured spindle axial and orthogonal tilts error motions. The calculated values of these error motions are 0.52 μm, 1.9 μrads and 1.9 μrads respectively.

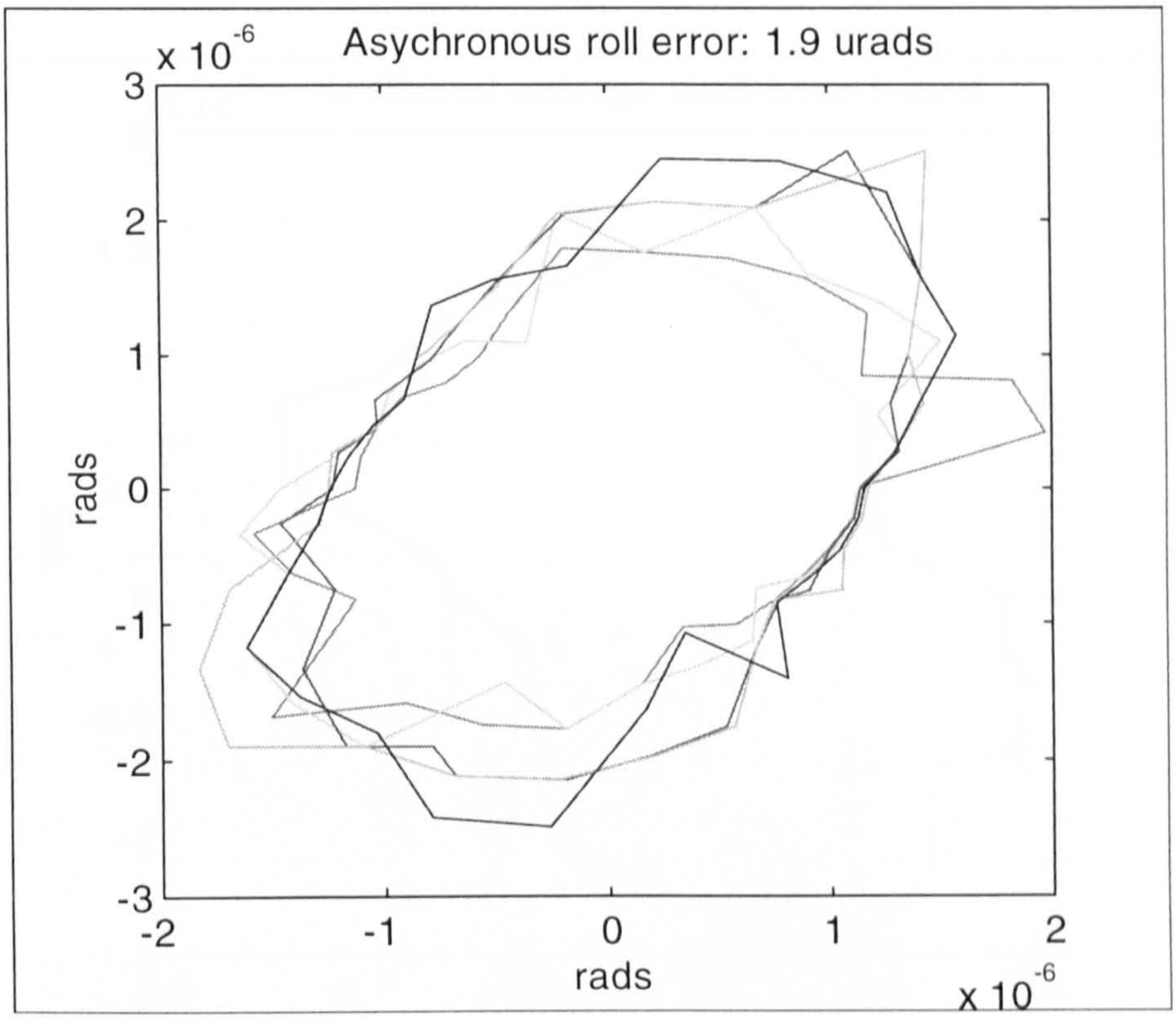


Figure 8.14c: Polar plot of spindle asynchronous Y-tilt error motion

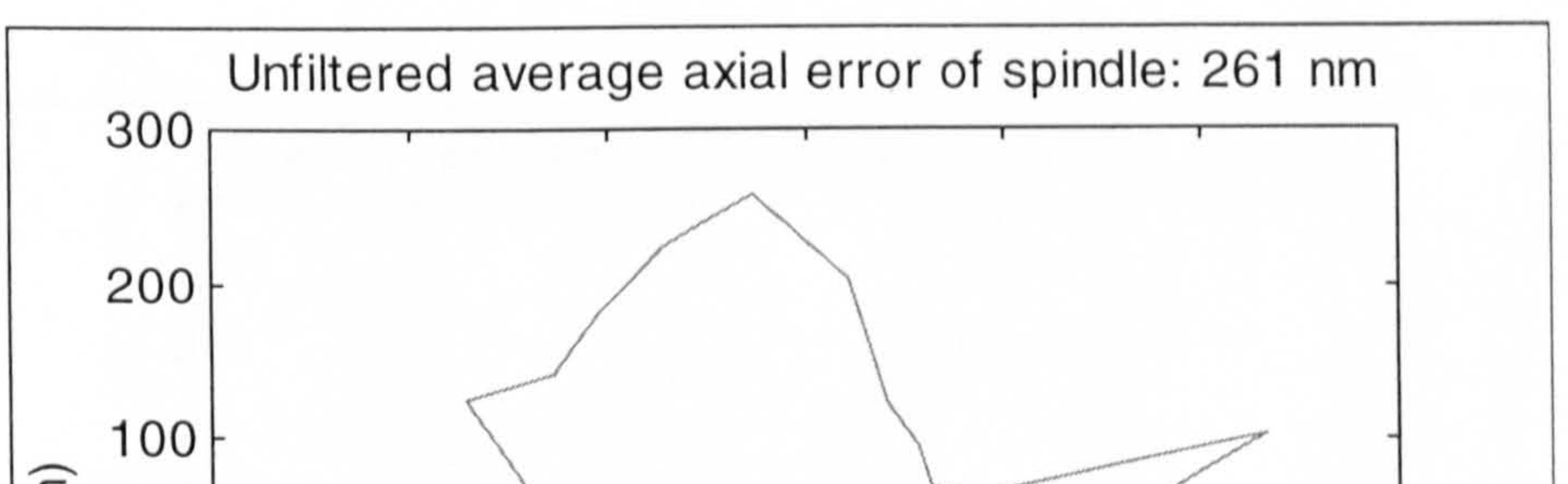


Figure 8.15a: Polar plot of spindle average axial error motion

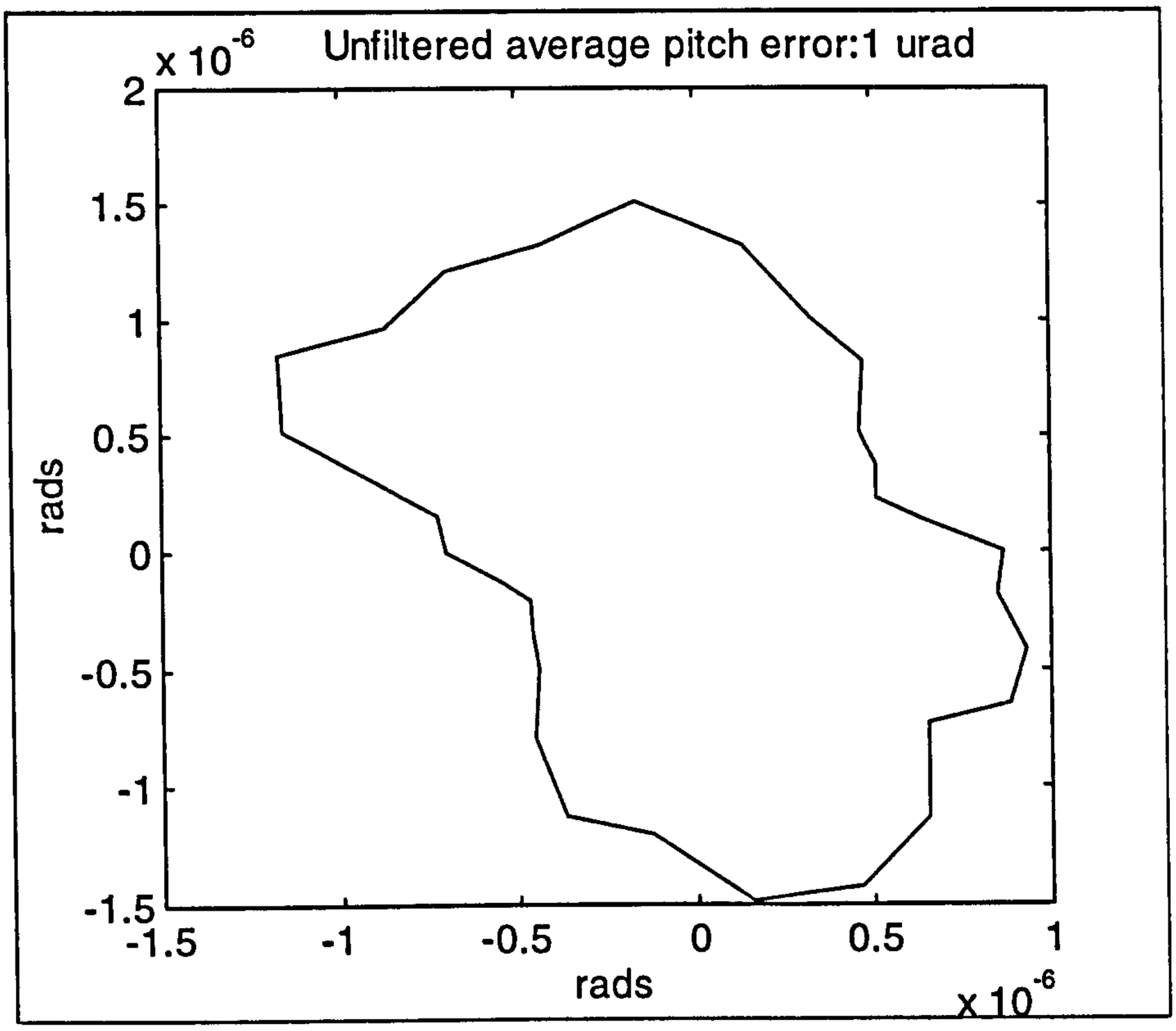


Figure 8.15b: Polar plot of spindle average X-tilt error motion

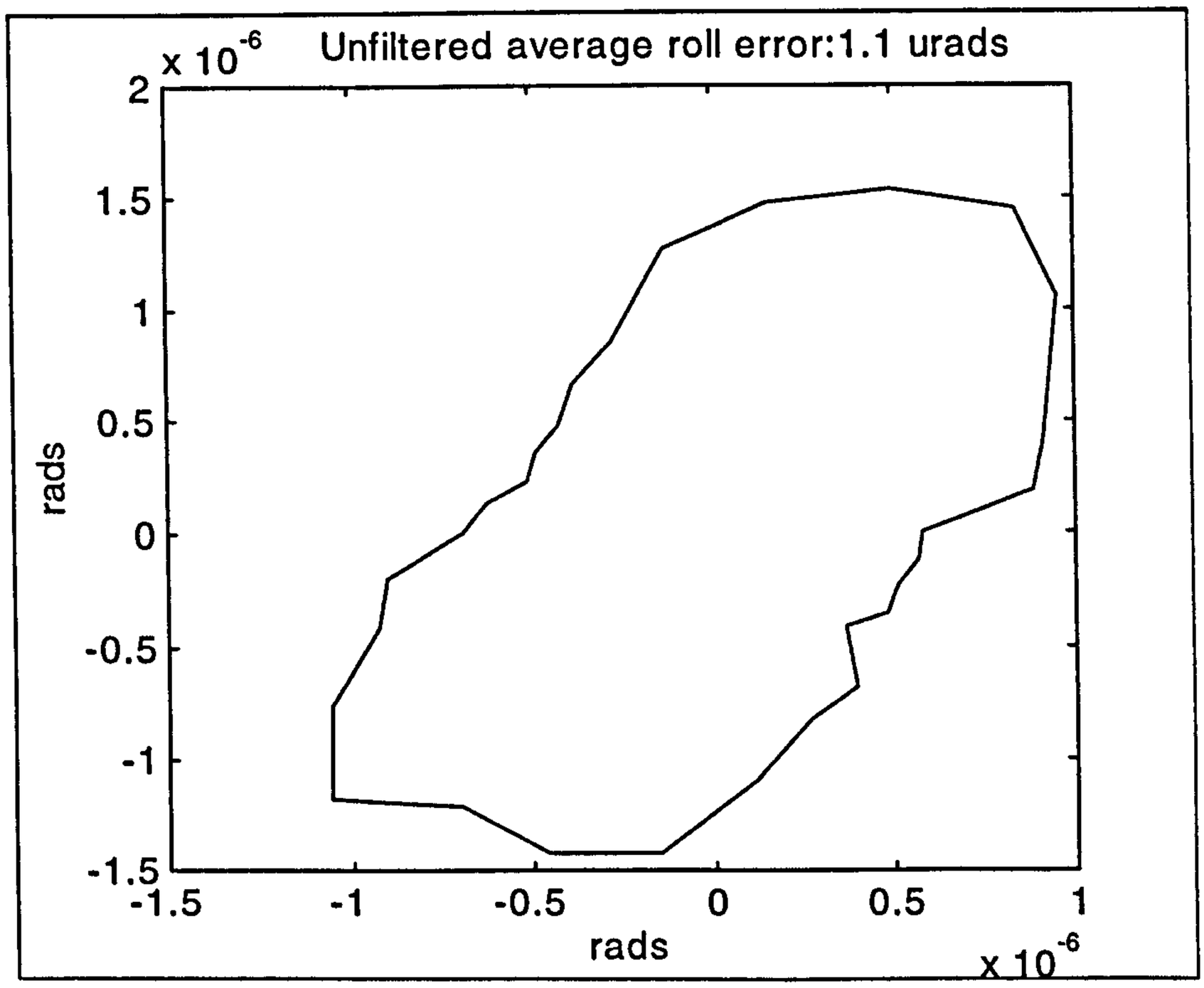


Figure 8.15c: Polar plot of spindle average Y-tilt error motion

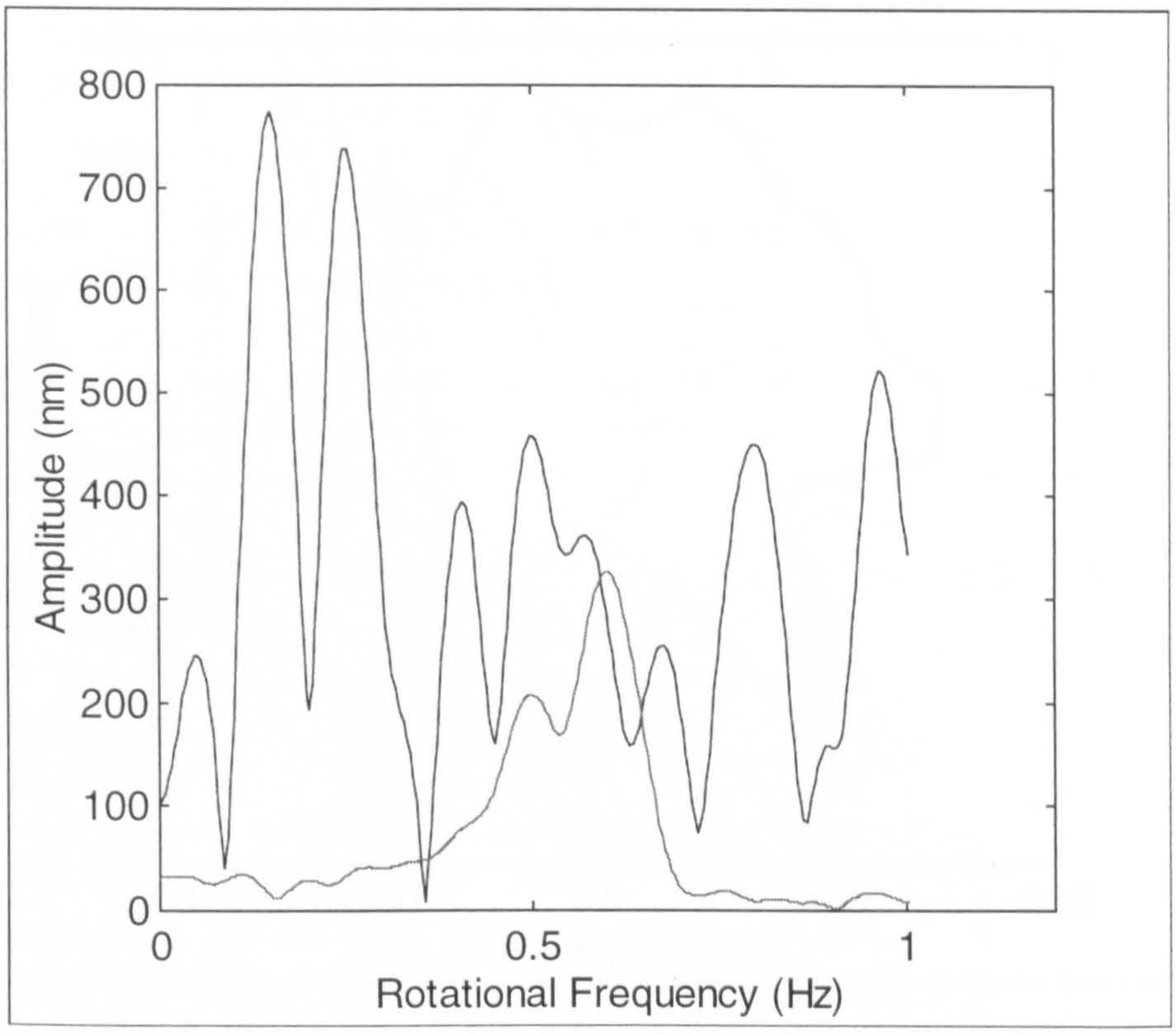


Figure 8.16: Frequency spectrum of average axial error motion

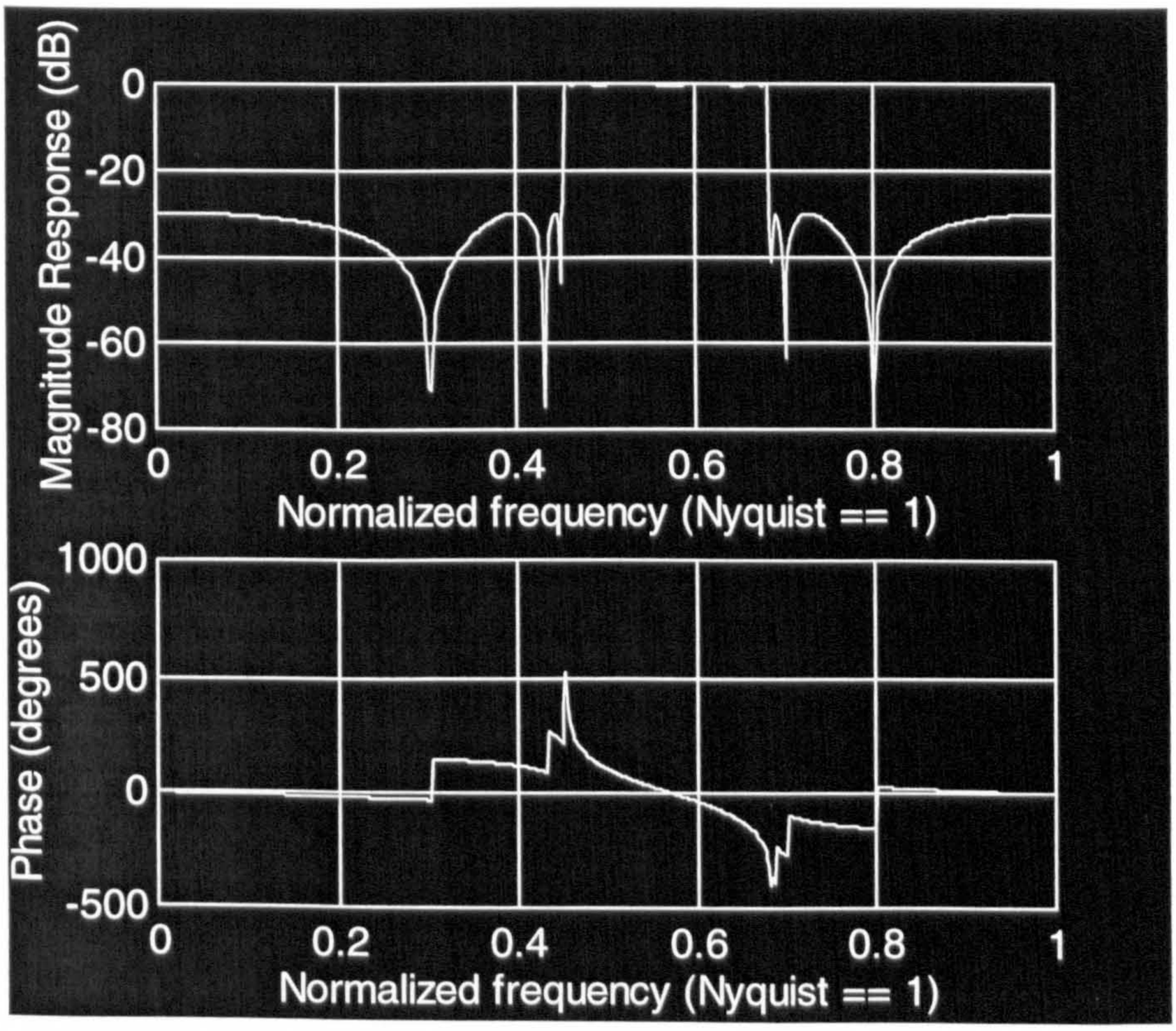


Figure 8.17: Phase and magnitude characteristics of band-pass filter

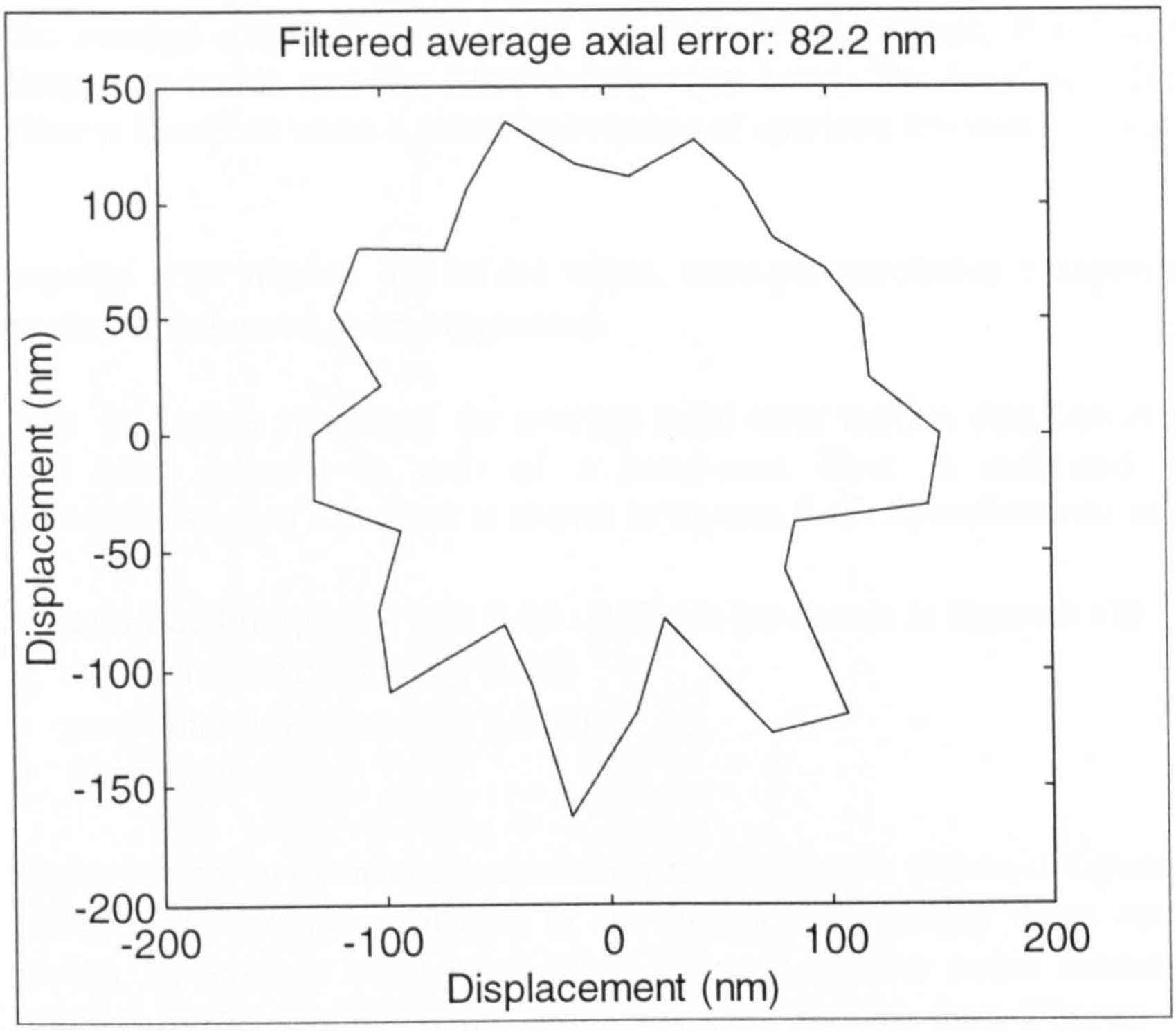


Figure 8.18: Polar plot of average axial error motion after filtering

The average error motions are the mean values of the total error motion. It is measured as the maximum and minimum radial difference of the error motion profile from the polar chart centre. Figures 8.15a, 8.15b and 8.15c show the average error motion polar plots of the interferometrically measured spindle axial and orthogonal tilt error motions. The calculated values of theses errors are 0.26 µm, 1 µrad and 1.1 µrad respectively.

8.5.4 NOISE REDUCTION OF ERROR MOTION DATA

Although the measured error motion values (shown graphically in figures 8.15a to 8.15c) are average values of the total error motion, they usually have high and low frequency noise components. A number of researchers in spindle metrology have used electronic or software based digital filters to remove these frequency components. For instance, Bryan et al (1975) used a low-pass electronic filter to eliminate high frequency components of the (synchronous) radial error motion values of a conical roller bearing. Slocum (1992 pp 543-550) explained the application of high-pass filtering for the spindle metrology data of a multi-axis grinder.

In this work, a band-pass elliptic (digital) filter (Ifeachor & Jervis, 1993) has been used to reduce the high/low frequency noise components of the average error motion values of the spindle. Figures 8.16, 8.19a and 8.19b show the frequency spectrum of

the average axial and orthogonal tilt error motion values. It indicates the high/low frequency bands and the filtered frequency band. The band-pass frequency of the filter is based on some a priori knowledge of spurious frequency components in the

average error motion (ie its d-c value, once-per revolution components and random peaks) which need to be suppressed.

The frequency spectra of the average axial error motion data before (shown in blue) and after (shown in red) of a band-pass filter is indicated in figure 8.16. Characteristics of this filter is shown in figures 8.17. Specifications of the filter are:

- pass-band frequency (f_p): 0.46 - 0.70 Hz (as shown in figure 8.18)
- stop-band deviation (d_s): 20 dB
- pass-band deviation (d_p): 2.3 dB
- filter order: 8

Improvement in the average axial error motion data is shown in figure 8.18, the figure indicates a minimal influence of the high/low frequency noise components, hence giving an average axial error of 82.2 nm. A similar noise reduction scheme was applied to the average X-tilt and Y-tilt error motion data. Figures 8.19a and 8.19b indicate the frequency spectra of the pitch and roll errors before (shown in blue) and after (shown in red) the application of a band-pass filtering. Specifications of this filter are:

- pass-band frequency (f_p): 0.39 - 0.58 Hz for average pitch error motion data
- pass-band frequency (f_p): 0.59 - 0.66 Hz for average roll error motion data

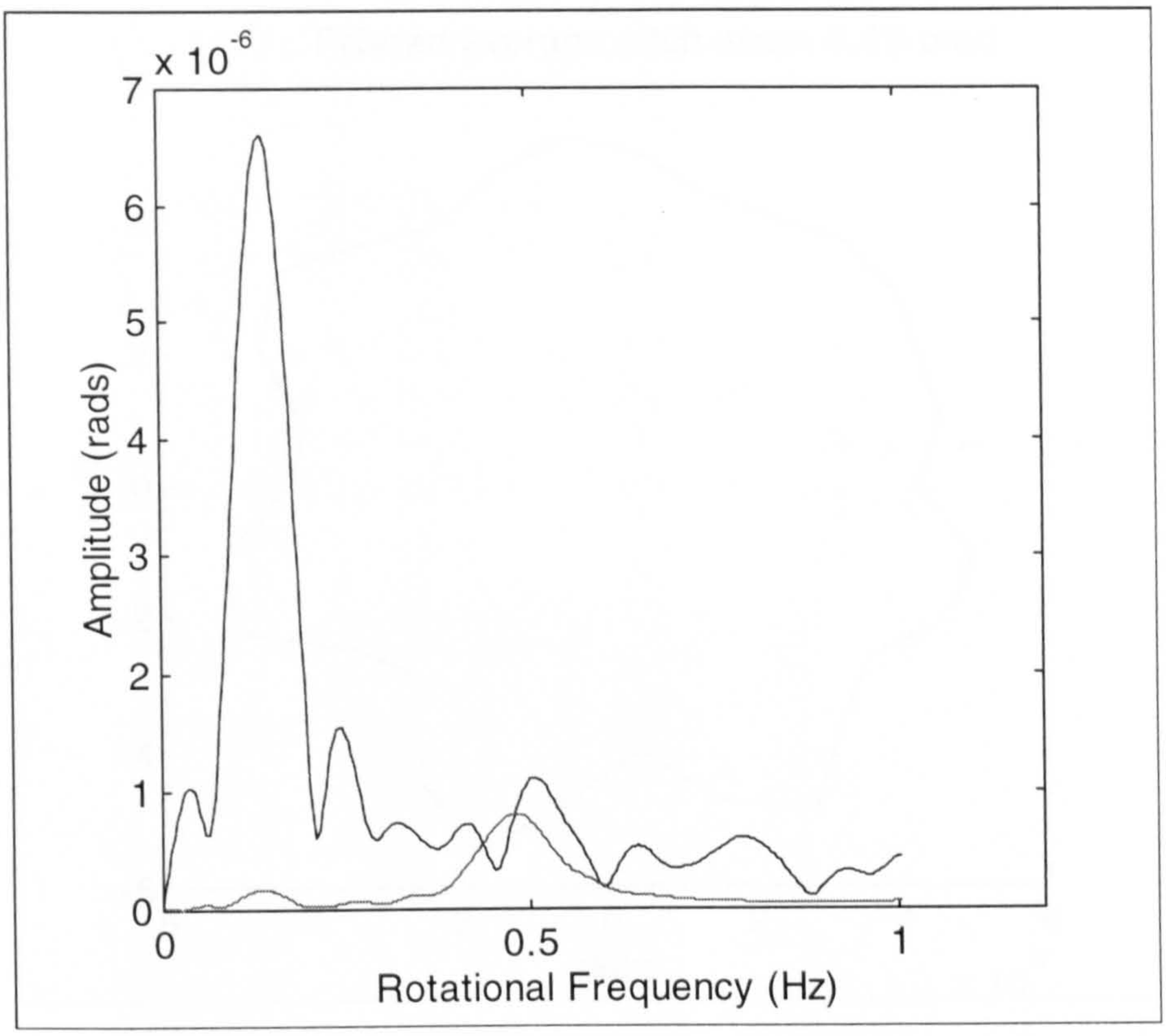


Figure 8.19a: Frequency spectrum of average X-tilt error motion

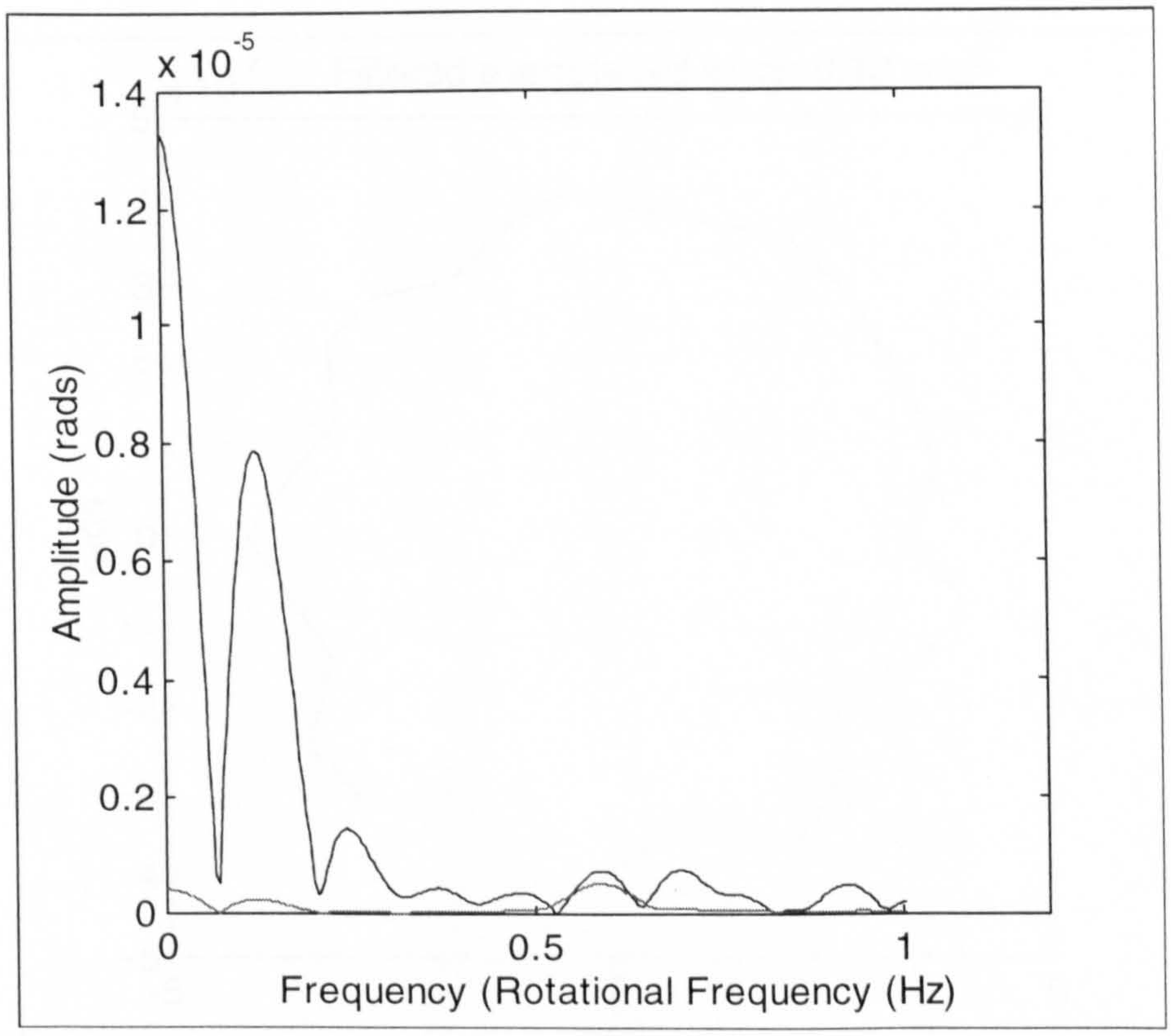


Figure 8.19b: Frequency spectrum of average Y-tilt error motion

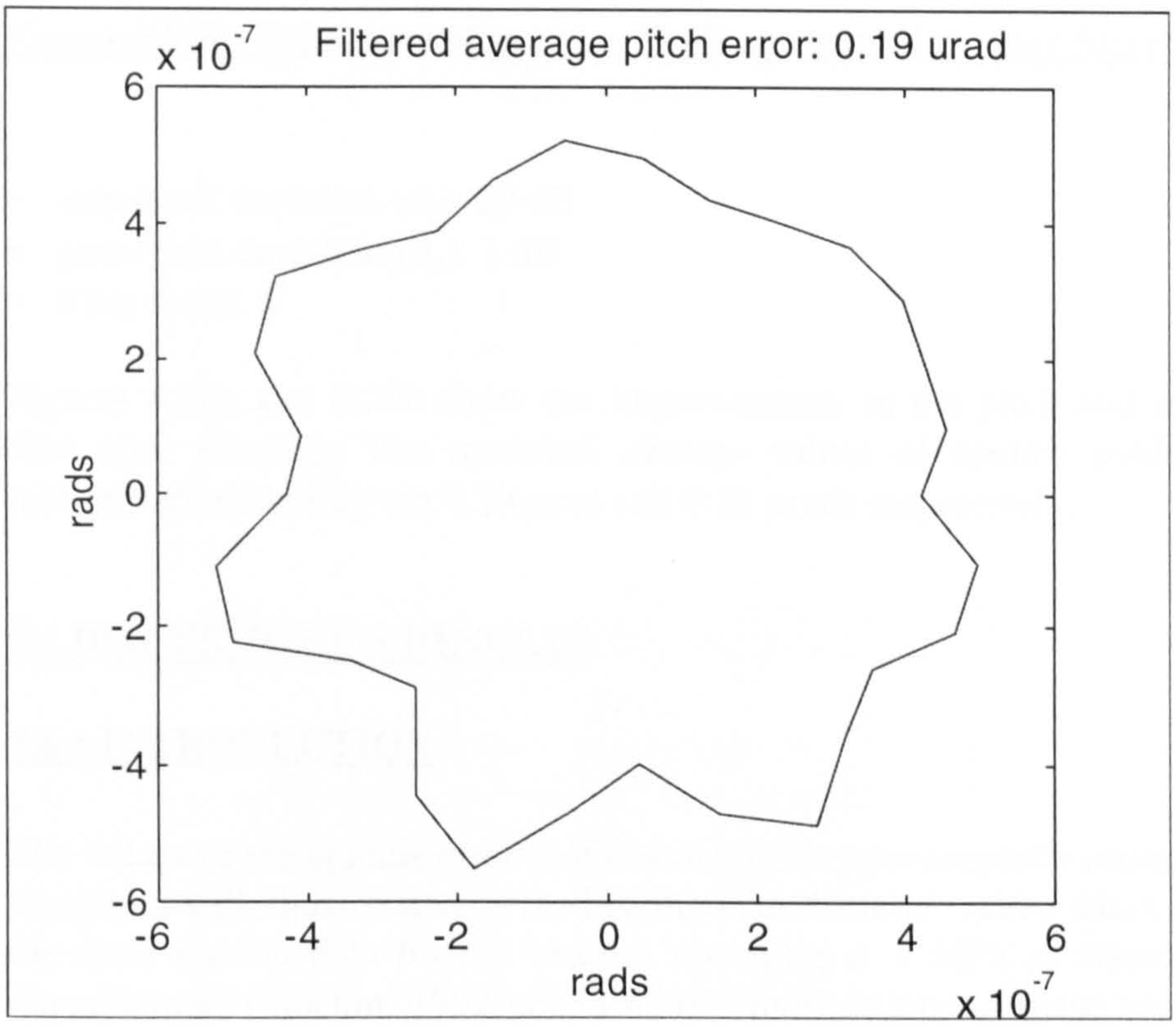


Figure 8.20a: Polar plot of average X-tilt error motion after filtering

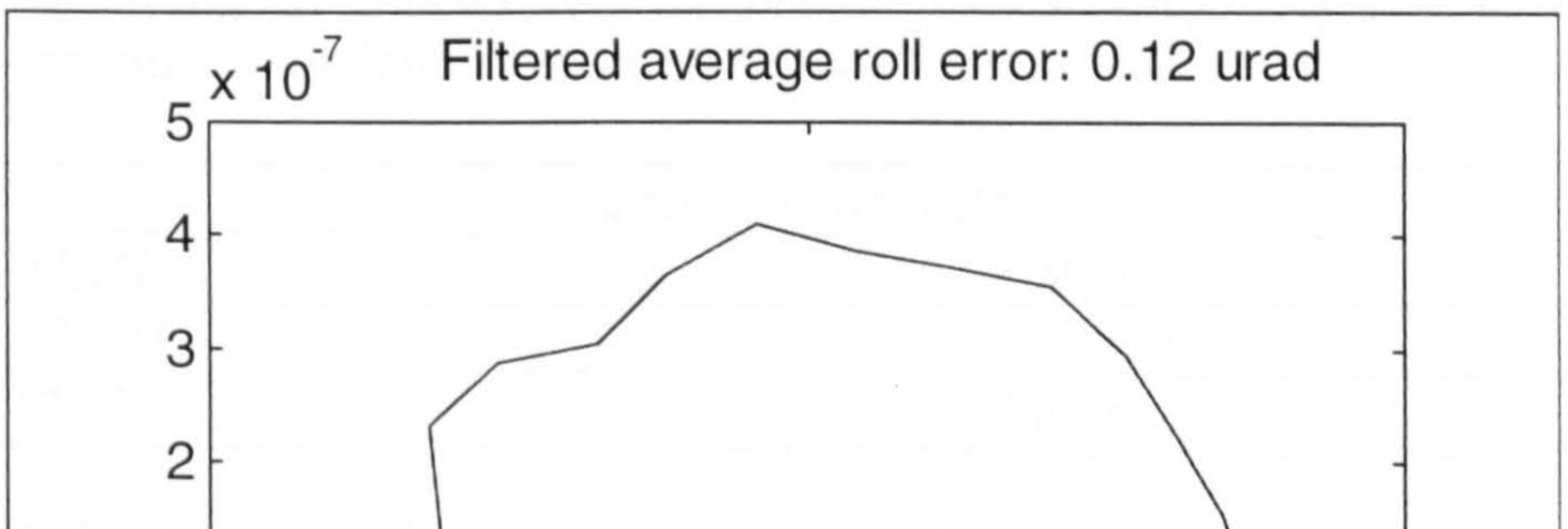


Figure 8.20b: Polar plot of average Y-tilt error motion after filtering

- stop-band deviation (d_s): 20 dB
- pass-band deviation (d_p): 1 dB
- filter order: 6

Figures 8.20a and 8.20b show the improvements in the pitch and roll error motion data after filtering. The modified average values of spindle pitch and roll error motions after filtering are 0.19 μrad and 0.12 μrads respectively.

8.6 DISCUSSION OF RESULTS

8.6.1 INTRODUCTION

The values of the spindle error motions obtained experimentally (*measured* data) were compared with those stated by the bearing manufacturer (*stated* data). Table 8.1 shows the error motion data for the bearing operating at 1 MPa as stated by the bearing manufacturer (Slocum, 1992 pp623-624). Modifications of these values are required in order to make the required comparison at the operating pressure of 0.6 MPa used in this work.

According to Slocum (1992, pp623-624), bearing load capacity, stiffness and compliance are (approximately) directly proportional to the operating pressure. Hence

ERROR MOTION	
RADIAL	< 50 nm
AXIAL	< 50 nm
TILT	0.2 μrad

Table 8.1: Spindle error motion data (obtained at 1 MPa operating pressure)

Type of error motion	***Measured* data obtained**	**Stated data (modified)**	**Deviation**
Axial	82.2 nm	< 83.3 nm	(with spec.)
Pitch	0.19 μrads	< 0.33 μrads	(with spec.)
Roll	0.12 μrads	< 0.33 μrads	(with spec.)

Table 8.2: Comparison between the measured and stated spindle error motion data (for 0.6 MPa operating pressure)

spindle error motions will vary inversely with the operating pressure (and bearing stiffness).

Table 8.2 shows a comparison of the *measured* and *stated* error motion values (modified for operating pressure of 0.6 MPa), it generally indicates very good correlation. However, there is a slight deviation between the *measured* and *stated* error motions as shown in table 8.2. In the following sub-section a number of factors which explain these deviations are described.

8.6.2 NUMBER OF DATA-POINTS (n_p) PER SAMPLE (s)

A high number of data points per sample (n_p) will give the better measurement resolution usually required to detect sub-harmonic error motion *lobes*. These lobes indicate the relationship between spindle error motions and the imperfect surface form in the bearing (thrust and journal) surface [ie these form errors cause changes in the forces (pressures) acting on the bearing surface at discrete angular locations]. In this work the value of n_p was 30, this was adequate enough to resolve the error motion lobes at an angular resolution of 6° as shown in figures 8.18, 8.20a and 8.20b.

The filtered axial error motion polar plot shown in figure 8.18 has 9 lobes and a peak-to-valley value of 96 nm. This result indicates the very high axial stiffness of the bearing typically about 1.8 kN/μm (Slocum, 1992 pp623-624), it is also indicative of the good surface finish and parallelism of bearing thrust surfaces.

The filtered angular (pitch and roll) error motion polar plots shown in figures 8.20a and 8.20b both have 7 lobes and their peak-to-valley values are 0.19 μrads and 0.12 μrads respectively. This indicates very accurate spindle alignment, due to the high angular stiffness provided by the bearing annular thrust and journal surfaces (Slocum, 1992 pp623-624). A typical value of the bearing angular stiffness is 11.76 N-m/μrads (Slocum, 1992 pp623-624). The angular accuracy also indicates good surface finish of bearing thrust and journal surfaces.

8.6.3 NUMBER OF SAMPLES (s)

A finite number of data-point samples (s) is required to obtain the average error motion data. A high value of s will give a better averaged error motion data. The value of s used in this work was 5, this gave a *mildly* averaged data which also exhibited some degree of randomness. The low value of s used in this work was due to the high computational requirements of each data-point (interferogram) ie each interferogram required a storage space of a 0.25 Mbytes and a processing time of 4 minutes per data-point (ie when processed on a 33 MHz 486 PC).

8.6.4 SELECTION OF FILTER BAND-PASS FREQUENCY

As mentioned in section 8.5.4 a band-pass filter was used to suppress components at d.c, very low frequencies and very high frequencies (random characteristics) of the averaged error motion data. These frequencies can be seen manifested in the error motion and frequency spectra plots given in figures 8.15a - 8.15c and 8.16, 8.19a - 8.19b receptively.

Adequate selection of band-pass frequencies is required in other to extract the relevant data related to spindle metrology. In this work this selection involved detecting the frequency band which gave a periodic but non-random average error motion plot as shown in figures 8.18, 8.20a and 8.20b.

8.6.5 VARIATION IN OPERATING PRESSURE

The error motion values evaluated in this work were obtained for the spindle operating at 0.6 MPa. In section 8.6.1, it was described that spindle error motion varies inversely with bearing stiffness. But according to Slocum (1992, pp623-624) this relationship is *approximately* a linear one, and as the operating pressure approaches atmospheric pressure (0.1 MPa) the relationship becomes highly non-linear. This implies that at the operating pressure of 0.6 MPa the relationship between bearing stiffness (error motion) and pressure is not as linear as that at 1 MPa (which is the operating pressure used by manufacturer to obtain the *stated* error motion values shown in table 8.2). This could partially contribute to the deviation between the *measured* and *stated* axial error motion.

CHAPTER 9: CONCLUSIONS & RECOMMENDATIONS FOR FUTURE WORK

9.1 INTRODUCTION

As described in the section 1.1, the need to measure spindle error motions to nanometric accuracy is required in a number of applications which include: machine tools, instruments and electro-mechanical devices. The work presented in this thesis concerned the development of a metrological system based on optical interferometry, for measuring spindle error motions in three degrees of freedom: axial-rectilinear displacement, and orthogonal angular. Results obtained in this work have shown the potential to interferometrically measure the spindle error motions to namometric levels of accuracy. In this chapter conclusions drawn from key areas of the thesis are presented. Some recommendations for the future direction of this work are also outlined.

9.2 CONCLUSIONS DRAWN FROM KEY AREAS OF THESIS

In section 3.3 a simple mathematical model was used to describe spindle error motions with the aid of homogeneous transformation matrixes (HTMs). The HTMs were used to define the spindle co-ordinate transformation from its reference position (ie its true axis of rotation) to a perturbed position.

In section 3.5, the dynamic characteristics of the spindle was modelled with the aid of mass (M), spring (K) and damper (C) components. The constant M relates to the inertia spindle rotor, while constants K and C relate to the stiffness and damping of the bearing and they both depend on the pressure gradient within bearing thrust and journal clearances.

The pressure gradient, in the bearing clearance volume can be describe analytically using Reynolds equation (see section 2.5.3). In section 3.5.4, it was mentioned that a number of researchers have developed numerical solutions (ie based on finite difference or finite element methods) of this equation were considered giving discrete values of pressure in the bearing which can be used to obtain the values of constants C and K. In this work the (axial , radial and angular) values of K were provided by the bearing manufacturer, while the values of C were evaluated analytically. Values of M, K and C, and resultant HTM of the spindle were used to define a set of error motion differential equations, as described in section 3.5.2. Solution of these equations gave a estimation of bearing radial, axial and tilt error motion values for specific spindle load capacity conditions. The estimated axial, radial and tilt error motions values were 19.4 nm, 9.7 nm and 0.19 μrad respectively. There were some slight discrepancies between the estimated and stated (ie by bearing manufacturer) values of the spindle error motion (as described in section 3.5.2). These discrepancies were due to the assumptions made in the spindle error motion estimation.

In chapter 3.5 the sources of spindle error motions were investigated, particularly for the combined thrust and journal bearing used in this work. Conclusions drawn from this investigation are that the sources of errors present in the bearing are due:

- bearing geometric inaccuracies [usually a major source of error motions (see section 3.5.2)]
- thermal effects [which affect bearing pressure distribution and bearing clearance (see section 3.5.3)]
- vibrational effects [due to forced or free bearing loading (see section 3.5.3)]

The bearing used in this work was manufactured to nanometric accuracy and the spindle structure was designed to exhibit high stiffness of typically 117 N μm^{-1}, 350 N μm^{-1} and 42 Nm μrad^{-1} in the radial, axial and angular directions respectively (Slocum, 1992 pp 623-624). The main source of error motion in this type of bearing is the out-of roundness of the rotor/stator and/or the supply restrictor pockets (Yabe, 1994a ,b).

In chapter 4, a number of techniques for measuring spindle error motion were described. These techniques were categorised as:

- capacitive
- optical diffraction
- backscatter fibre-optic
- Moiré optic
- inductive
- optical interferometry

It was shown in section 4.2 that the capacitive technique (ie a commonly used method for measuring spindle run-out) has a very high measurement bandwidth and resolution, but can be influenced by cross-talk between sensor electrostatic flux lines and when two or more probes are used simultaneously.

It was demonstrated in section 4.3 that the phenomenon of Fraunhofer diffraction can be used to measure spindle run-out by detecting the variation in the angular half width of a slit diffraction sinc intensity function. The technique was shown to have a nanometric resolution (see figure 4.30) , but issues such as positional accuracy of the slit formed between a knife edge and spindle target surface (ie which can be influenced by vibration) need to be addressed.

Fibre-optic displacement sensing (described in section 4.4) can be employed to work on the principle of intensity modulation of a light reflected from the displaced spindle target surface. The viability of this technique is subject to the surface finish of the target surface, and it was shown to be suitable for sub-micrometre spindle metrology.

In sections 4.5.4 and 4.5.5, inductive techniques for measuring spindle run-out were described, in particular as linear variable differential transformer (LVDT) and eddy current sensing. LVDT sensors are adequate for characterising the performance of hydrodynamic or *metal-to-metal* contact (ball or roller) spindle bearings to sub-

micrometre accuracy. However it is inappropriate for the run-out measurement of ultra-precision (aerostatic or hydrostatic) spindles due to its invasive nature. Eddy current sensors on the other hand are non-invasive and have a sub-namometric measurement resolution, making them suitable for the run-out measurement of ultra-precision spindles.

Moiré and interferometric sensing (described in sections 4.7 and 4.6) have some similarities, since both techniques require fringe analysis for the acquisition of error motion data. In section 4.6, it was outlined that Kim and Park (1993) developed a Moiré technique (based on circular gratings) for measuring spindle radial run-out with a measurement resolution of typically 10 nm.

The interferometric sensing technique developed in this work uses an intensity-based fringe analysis approach, which gives a measurement resolution less than a 40th of a wavelength [ie by increasing interferogram spatial resolution (section 8.3.2) and by using sub-pixel interpolation (section 6.5)] . However, it was mentioned in section 4.7.5, that the configuration of the interferometer used in this work (ie which is a Fizeau interferometer) is prone to environmental effects such as the changes in pressure, humidity and temperature which inherently affect the refractive index and reduce the accuracy of the instrument. These effects can be reduced by using a common-path interferometer or by measuring the changes in refractive index in-situ with a refractometer.

A summary of some of the benefits of the interferometric sensor developed in this project are:

- good fringe peak measurement resolution to less than 6 nm (by using the fringe interpolation technique outlined in section 6.5)
- full field technique (single sensor required to measure 3 DOF error motions)
- insensitive to electrical noise
- measurement range is adequate for measuring spindle error motions
- non-invasive

In section 5.3.2, the requirements for interferometric measurement of spindle run-out and real-time detection of rotating interferograms were outlined. One of the main issues to be considered during interferogram acquisition, was how to capture and store rotating fringe pattern data over a period of duration shorter than changes occurring in the inteferogram due to spindle error motions.

In section 5.3.2, detection of the interferogram data was categorised as:

- *synchronous detection*: where interferograms are captured at specified angular positions in one spindle revolution

- *asynchronous detection*: where interferograms are captured at specified angular positions over a number of spindle revolutions

One of the problems of synchronous and asynchronous detection of the fringe pattern at the standard video frame-rate of 25 frames/s (40 ms) for a spindle rotating at high speed (ie of the same cyclic order) is aliasing. This problem causes the interferogram to appear swivelled. The swivelled interferogram is due to the make-up of the observed frame, which is composed of *sub-frames* captured as the spindle rotated through discrete angular positions during the 40 ms frame capture time.

This problem was solved by sampling the interferograms synchronously at low spindle speeds (ie approximately at 60 rev/min). This allowed the camera to have sufficient time to sample a full interferogram frame at the standard rate of 25 frames/s.

Chapter 6, describes an *intensity-based* interferogram analysis technique developed to analyse fringe patterns generated by the spindle interferometer. This technique (described in section 6.2) was based on *a priori* knowledge of the generated fringe pattern intensity profile and was used to detect interferogram orientation, fringe density, fringe spatial frequency and fringe phase. This technique relies on the measurement of fringe peak positions and intensity values in a defined *area of interest* of a sampled interferogram. Compared with other interferogram analysis techniques, such as temporal-phase and spatial-phase, the intensity-based technique is relatively easy and not computationally demanding.

The intensity-based interferogram analysis technique used in this work incorporated a number of pre-processing algorithms (see section 6.4) such as optimal thresholding, edge enhancement, morphological functions, geometric transformation, spatial and Fourier filtering. These algorithms were used to enhance fringe pattern morphology, edge and intensity distribution which are nominally affected by the influence of noise. The Fourier transform approach was used to detect the fundamental frequencies of interferograms. This proved to be a viable technique for detecting fringe orientation, spatial frequency and spatial phase.

The measurement of fringe peak positions of the acquired interferograms was further improved by incorporating a sub-pixel peak detection algorithm (as outlined in section 6.5. This involved fitting a polynomial function to neighbouring values of fringe peak intensity data, and it improved the fringe peak position detection to sub-pixel accuracy (ie with root-mean-square interpolation error of less than a pixel).

Section 7.1, considered adjustment of the interferometer for spindle metrology. A two-axis servo (described in section 5.5.1) was used to automatically adjust the interferometer as the spindle rotated. A linear model (described in section 7.5.2.2), which related servo input voltage (e_r) with fringe density (p), was used to describe the characteristics of the two-axis servo. Interferometer adjustment was achieved by two feed-back control regimes:

- on-off control (with and without dither)
- model reference adaptive control (MRAC)

The issue of servo non-linearity error (considered in section 7.6) was investigated in this project. A technique known as *dithering* (described in 7.7.2) was used to reduce this error from 1.1 to 0.86 fringes and 1.70 to 1.07 in the X and Y axis servo

respectively. The dither technique was used to suppress servo backlash non-linearity error, while the remaining non-linearity errors such as stochastic noise and fringe measurement errors were minimised by other means. In section 7.7.3, it was shown that the MRAC approach essentially negates the effect of backlash and stochastic errors during servo control since the *physical* and *reference* outputs correlate very well.

The interferometric measurement system developed in this project was used to measure spindle axial (d_z), X-axis tilt (E_x) and Y-axis tilt (E_y) error motions. In section 8.1 it was outlined that these error motions were calculated from interferogram spatial and Fourier transform data. Immediately before the spindle error motion measurement, interferograms are checked for the condition of constant fringe density (or spatial frequency) using the adjustment routine described in figure 8.1.

Spindle error measurements were obtained for 5 batches (samples) of 30 data-points, sufficient to obtain the required axial and tilt error motion polar charts as described in section 8.5.3. This data subsequently was averaged giving axial, and orthogonal tilt error motions of 0.261 μm, 1.0 μrad and 1.1 μrad respectively. In section 8.5.4, a band-pass filter was used to suppress the d.c, very-low frequencies and some of the random noise of the average error motion data. This scheme reduced the spindle axial, and (X and Y) tilt average error motions by 0.179 μm, 0.81 μrad and 0.98 μrad respectively.

The values of spindle error motions obtained in this work (ie referred to as the *measured* values in section 8.6.1) have been compared with those stated by the bearing manufacturer (ie referred to as the *stated* values). The comparison between the two indicate very good correlation as shown in table 8.1. However there is a slight discrepancy between the measured and stated error motions, which in section 8.6.1 was attributed to:

- number of data-points per sample
- number of samples
- selection of filter band-pass frequency
- selection of bearing supply pressure

It has been noticed recently (on the internet) that the error motions values *stated* by the bearing manufacturer have been improved (see attached data sheets obtained from the bearing manufacturer web site). Table 9.1 shows these modified values, obtained for a bearing operating at 1 MPa and 0.6 MPa respectively.

Error motion	Supply pressure: 1 MPa	* Supply pressure: 0.6 MPa
Axial	25 nm	41.67 nm
X-tilt	0.05 µrads	0.08 µrads
Y-tilt	0.05 µrads	0.08 µrads

Table 9.1: Recent modified values of spindle bearing

It is speculated that the reasons for these developments are:

- improved accuracy in bearing manufacture
- improvement in bearing error motion measurement
- better quantity control (ie statistical process control)

9.3 RECOMMENDATIONS FOR FUTURE WORK

9.3.1 INTRODUCTION

In this section a number of recommendations are proposed for the future improvement of the spindle metrology system developed in this work. Particular areas of the system which warrant future improvement include:

- interferogram acquisition rate
- interferogram analysis
- interferometer configuration
- system modification for measurement of spindle radial error motions

9.3.2 HIGH-SPEED INTERFEROGRAM ACQUISITION

In section 5.3.3 it was pointed out that a high-speed low resolution CCD camera could be used to increase the interferogram acquisition rate. A frame rate of 25 frames/s was used in this work for the measurement of a spindle revolving at 60 rpm. In order to measure spindle error motions at high speeds (ie typically up to 2000 - 3000 rpm), it was estimated in section 5.3.3.2 (ie using data given in table 5.4) that a CCD camera operating at 2,900 frames/s would be required.

---

* The values have been corrected to match the operating supply pressure condition used in this work (ie based on a linear relationship between bearing stiffness and operating pressure as described in section 8.6.1)

One of the consequences of using a high frame rate camera of the type described in section 5.3.3, is that its spatial resolution decreases as its frame rate increases. For instance the required 2,900 frames/s CCD camera has a spatial resolution of 64 × 64 pixels compared to 512 × 512 pixels provided by the Vidicon camera used in the project. This is because a camera with a 64 × 64 spatial resolution has few photodetectors than a 512 × 512 camera, and thus requires a much shorter exposure time for a full frame. This problem can be resolved by increasing the resolution of the interferograms prior to their acquisition, by modifying the optics of the spindle interferometer (ie by fitting a *macro zoom* lens facility onto the interferometer camera)
.

9.3.3 HIGH-SPEED INTERFEROGRAM ANALYSIS

A number of steps such as spatial filtering, binarisation, morphology, edge enhancement were used to process the interferogram data (see section 6.4 for more details). It was pointed out in section 8.6.3 that each interferogram took approximately 4 minutes to process on a 33 MHz 486 personal computer. As a result of this it took a total of 10 hours to process 150 interferograms to generate the error motion polar plots described in section 8.5.2

It is proposed that in the future a high speed *image processing card* should be used to reduce the current interferogram processing time. An image processing card generally incorporates a number of application specific integrated circuit (ASIC) digital signal processing chips which provide high speed image acquisition and real-time image processing functionality. A modern PC might be expected to have a 333 MHz processor and when used with an image processing card, can improve the current interferogram processing time quite considerably.

Theses cards can be purchased from a number of suppliers such as EPIX*, CORECO** and ALRAD***. Some popularly known image processing cards are the OCULUS F/64 and 4MEG VIDEO supplied by CORECO and EPIX respectively.

One of the important features of an image processing card is that its processing and memory architecture is designed for simultaneous image acquisition and processing. This dual functionality makes it possible for images or interferograms to be analysed in less than one frame time. It also has a high speed memory communication bandwidth which provides access to previously stored data even while interferogram acquisition is in progress (DALSA, 1995). Although an image processing card is usually interfaced to a high-speed personal computer, it works autonomously and hence its data processing is independent of the personal computer processor.

---

* EPIX, Inc., 381, Lexington Drive, Buffalo Grove, IL 60089
** CORECO, 6969 Trans Canada Highway, Suite 142, St, Laurent, PQ, H4T 1V8
*** ALRAD Instruments Ltd., Alder House, Turnpike Industrial Estate, Newbury, Berkshire, RG14 2NS

In general, an image processing card is a parallel processor dedicated to the high speed processing of images (or interferograms), they have an estimated processing time of 12 million instructions per second.

9.3.4 INTERFEROMETER CONFIGURATION

In section 4.7.5.1, it was described that the Fizeau interferometer used in this work had a *dead-path* of approximately 300 mm between the reference and test surfaces. The dead-path of the interferometer is sensitive to environmental changes such as variations in temperature, pressure and humidity, which introduces errors in refractive index and optical path length.

A solution to this problem is to change the interferometer set-up to a configuration with a zero dead-path such as a Mach-Zender, Michelson, Williams or common-path interferometer (see for example, Dyson, 1970).

9.3.5 MEASUREMENT OF RADIAL ERROR MOTIONS

9.3.5.1 INTRODUCTION

The spindle metrology system developed in this work was used to measure spindle axial and tilt error motions ie three degrees of freedom. In order to measure spindle radial error motions (two further degrees of freedom), the current interferometric system needs to be extended. In this section a number of ways of modifying the system to measure spindle radial error motions are proposed.

9.3.5.2 HYBRID INTERFEROMETRIC AND CAPACITIVE (OR INDUCTIVE)

A hybrid interferometric and capacitive (or inductive) technique can be used to measure the five degrees of freedom (DOF) spindle error motions simultaneously. Such a set-up would exploit the interferometric approach developed in this project for the measurement of spindle axial, pitch and roll error motions, and incorporate two capacitive (or inductive) probes for the measurement of spindle orthogonal radial error motions.

The capacitive (or inductive) probes would be positioned radially on the periphery of the spindle at right angles to each other as shown in figure 9.1. It was mentioned in sections 4.2 and 4.5.5 that both capacitive and inductive sensing have the capability for sub-micrometre to nanometre resolution, adequate for the ultra-precision spindle metrology.

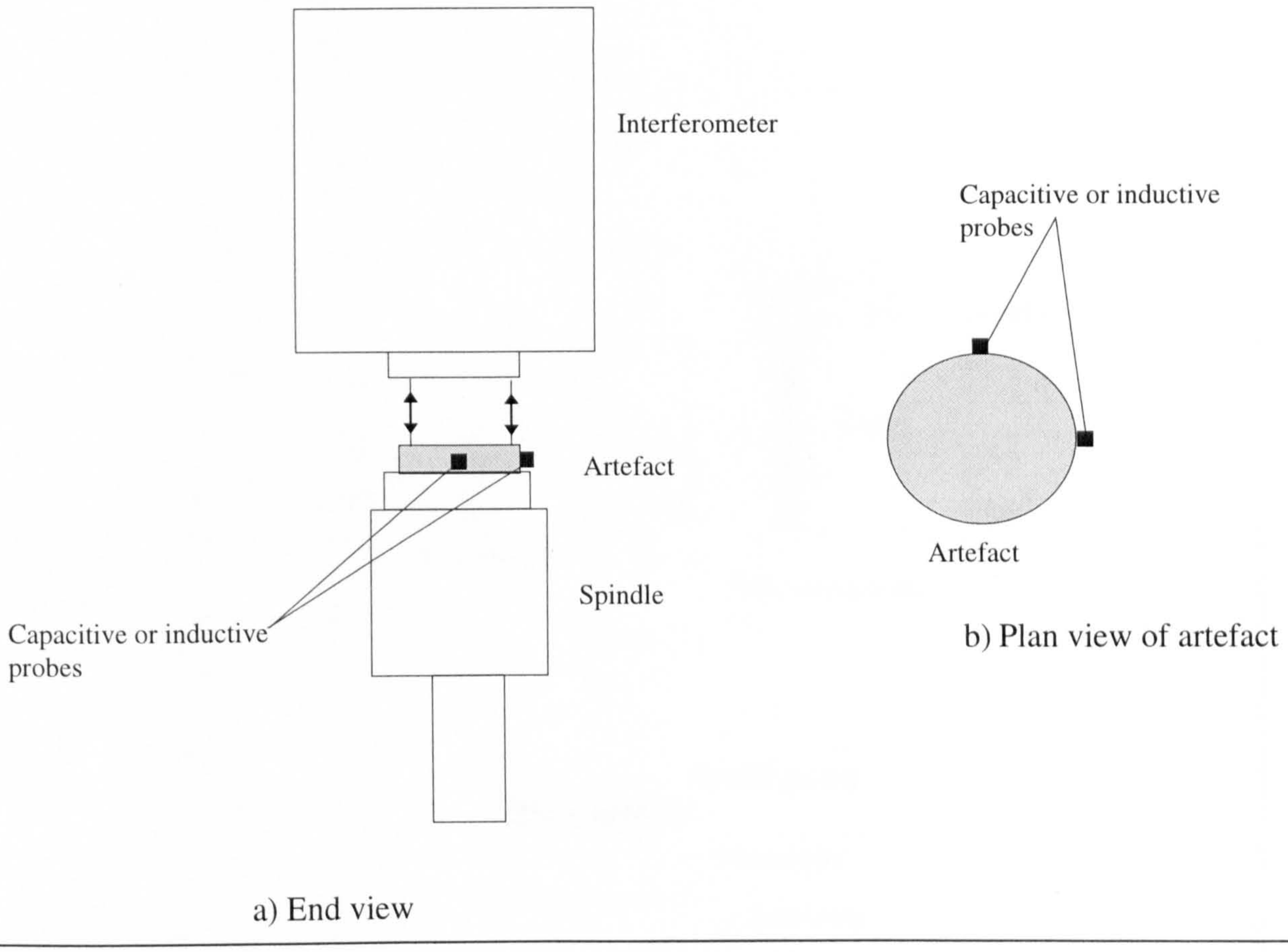


Figure 9.1: Schematic showing the end and plan view of hybrid interferometric and capacitive (or inductive) technique

The hybrid interferometric and capacitive (or inductive) technique has the benefit of noise/interference immunity and whole field measurement capability provided by interferometry, and good measurement resolution and frequency response provided by capacitive (or inductive) sensing. It also has the potential for cross-compensation of measurement techniques.

9.3.5.3 HYBRID INTERFEROMETRIC AND MOIRÉ MEASUREMENT

In section 4.6.2, the Moiré technique was considered for spindle metrology. Kim and Park (1994) show that by mounting a circular grating onto a spindle face-plate seen through a concentric stationary grating, fringes are formed and the variations of these fringes during rotation can be used to measure spindle radial error motions.

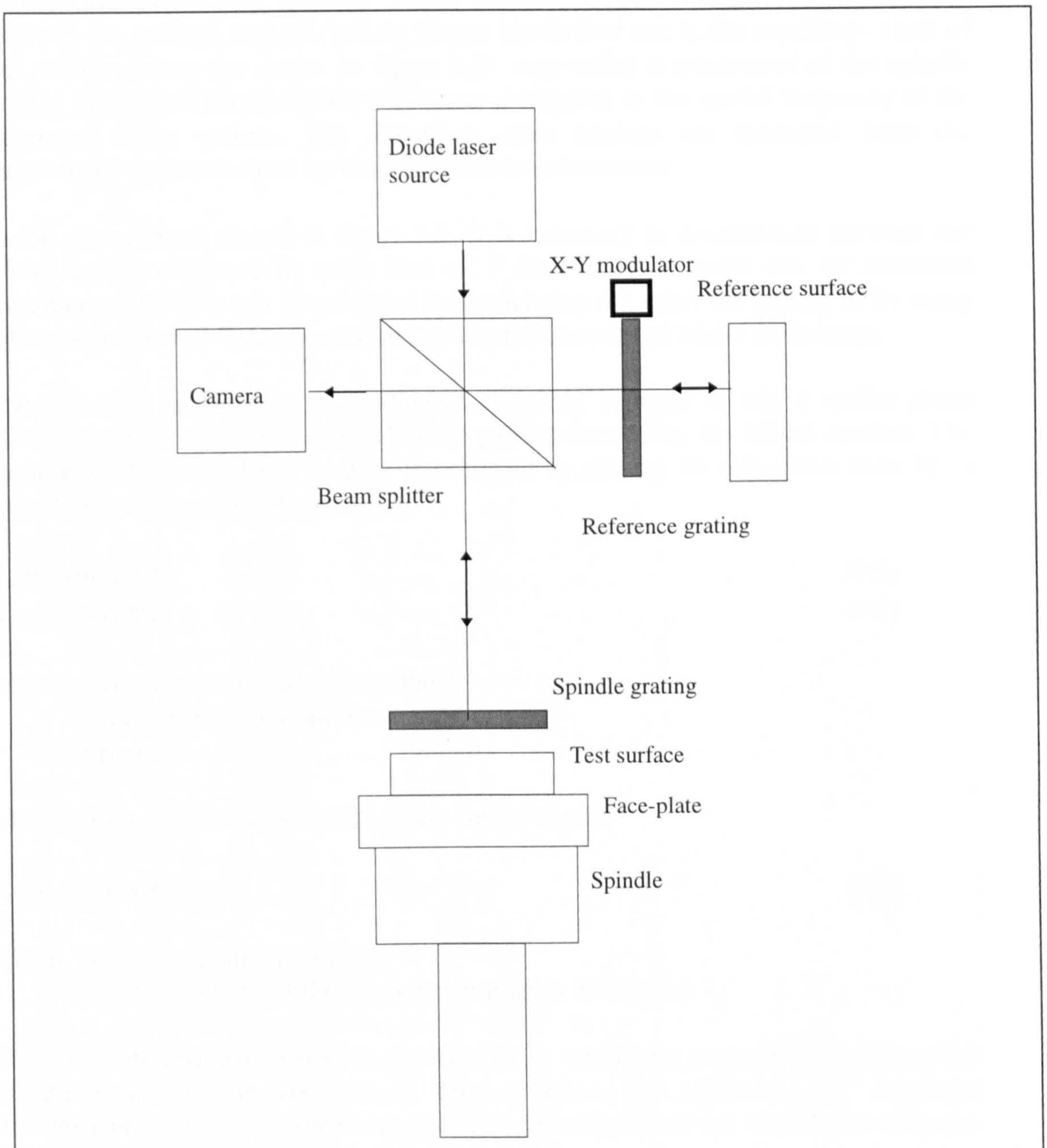


Figure 9.2: Schematic of hybrid interferomertric and Moiré technique

There are a number of different grating formats (circular, radial and spiral) which can be used for metrological applications (Dyson, 1958). A Moiré technique which uses two circular gratings is proposed to measure the 2 DOF spindle radial error motions, and an interferometric approach (similar to that developed in this work) to measure the 3 DOF spindle axial and tilt roll error motions, giving a total of 5 DOF error motion measurements.

A schematic of the hybrid interferometric and Moiré set-up is shown in figure 9.2. The figure shows a Michelson interferometer, a camera, spindle test surface and two gratings (reference and spindle). When there is a relative coplanar displacement between the gratings straight (radial) fringes are formed due to the superimposition of the grating pattern (as shown in figure 9.3). Any radial displacement of the spindle grating (ie the two further DOF) will cause a variation in the spatial frequency of the generated fringe pattern. The remaining error motions are measured from the interference pattern formed by the Michelson interferometer.

In the arrangement shown in figure 9.2, it is necessary to discriminate between the Moiré and interference in order that all 5 DOF error motions can be measured simultaneously. This can be achieved by modulating the reference grating or by using different (coloured) light sources for the interferometric and Moiré techniques.

The reference grating can be modulated circularly in order to add a spatial phase carrier signal to the nominal interference pattern formed by the Moiré method. The circular modulation of the grating is achieved by moving its orthogonal axes by a pair of sine and cosine signals:

$$\Delta_x = r \sin(2\pi ft) \tag{9.1}$$
$$\Delta_y = r \cos(2\pi ft) \tag{9.2}$$

where, r: amplitude of modulating signal
f: modulating temporal frequency
t: time

which produces a resultant spatial phase carrier signal:

$$\Delta_o(\Delta_x, \Delta_y) = f_0 S(x,y) \tag{9.3}$$

where, f_0: carrier spatial frequency
S(x, y): position of the reference grating in X-Y space

The modulating signal defined in equation (9.3) introduces a spatial frequency offset in the resultant intensity pattern formed when the reference and measured transmission gratings are superimposed. The trasmittance of the modulated reference and spindle gratings of equal pitch (p) are:

$$T_r = a + a\cos 2\pi\left(\frac{r(x,y)}{p} + \Delta_0\right) \tag{9.4}$$

$$T_m = a + a\cos 2\pi\left(\frac{r(x,y)}{p} + \Delta_m\right) \qquad (9.5)$$

and according to Gasvik (1995, pp163), when they interfere the resultant Moiré intensity pattern can be defined mathematically as:

$$I_r(x,y) \propto T_r T_m = I_0 + I_1 \cos 2\pi(\Delta_0 + \Delta m) + F \qquad (9.6)$$

where, I_o: mean intensity of light source
I_1: AC term of the intensity pattern
Δ_m: measured phase due spindle radial error motions
F: higher spatial frequency terms

Equation (9.6) indicates the presence of a spatial phase carrier (Δ_o), which is manifested as a finite number of fringes detected by the camera, this is added vectorially to the nominal number of fringes formed due to the spindle error motion. When there is no spindle radial error motion (ie $\Delta_m = 0$), a fringe pattern of constant spatial frequency is detected, this fringe pattern is due to modulation of the reference grating. In order to avoid any sampling errors during the acquisition of the Moiré fringe pattern, the modulation frequency of the spatial phase carrier must be greater than the maximum rotational frequency of the measured spindle (ie the carrier frequency must be faster than measured phased due to spindle radial error motions).

According to Takeda et al (1982) equation (6) can be transformed to the frequency Fourier domain:

$$A(x,y) + C(x,y)(f_0 - f_m) + C^*(x,y)(f_0 - f_m) \qquad (9.7)$$

where, A(x,y): mean intensity in Fourier space
C*(x,y), C(x,y): *real* and *imaginary* intensity AC terms in Fourier space
f_0, f_m: carrier and measured spatial frequencies

Introduction of the carrier spatial frequency f_o enables the Moiré fringes to discriminated from the interferometric fringes in the frequency domain by using the fast Fourier transform filtering technique described in section 8.3.3 (see figure 9.4)

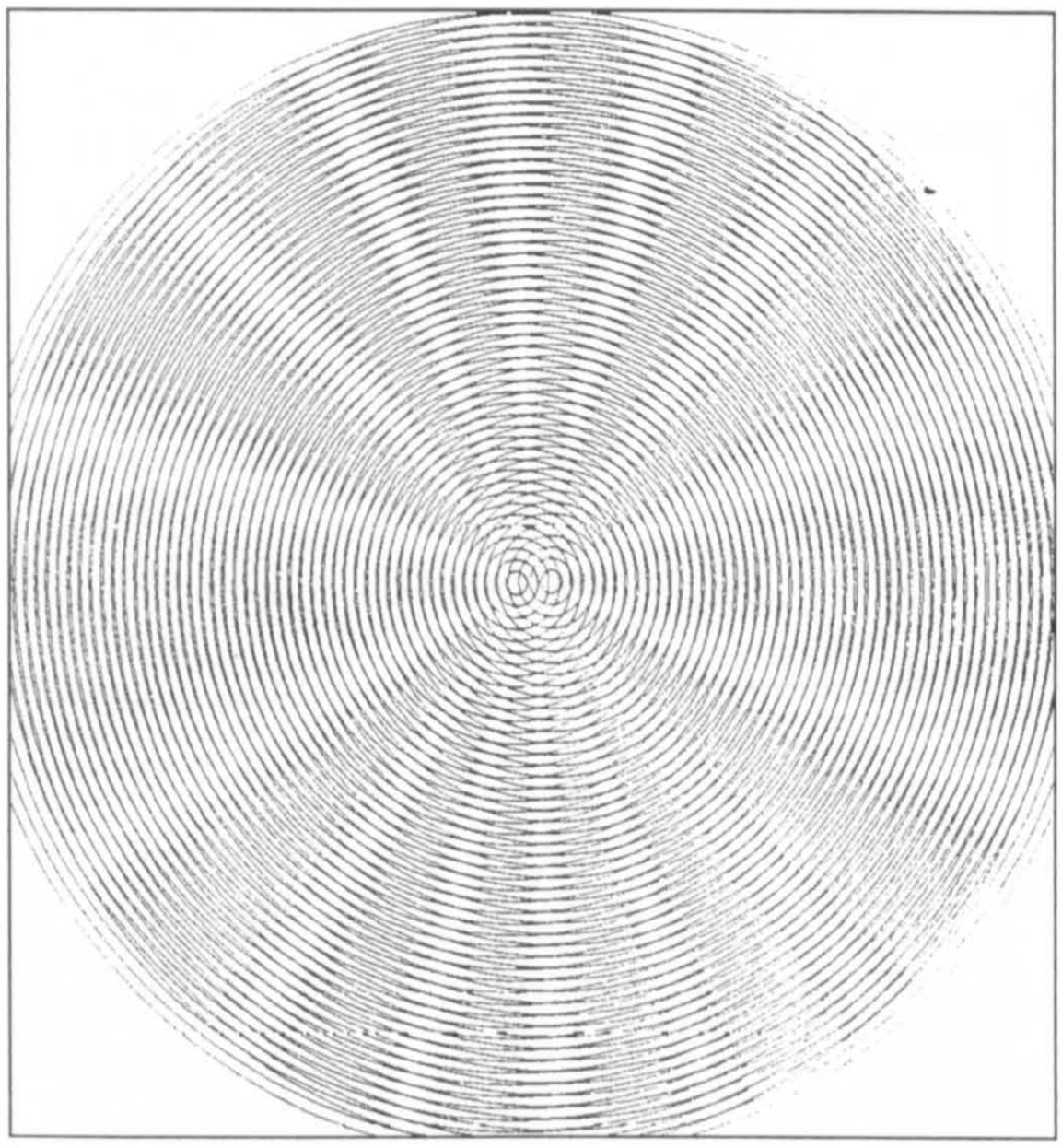


Figure 9.3: Moiré pattern of two circular gratings

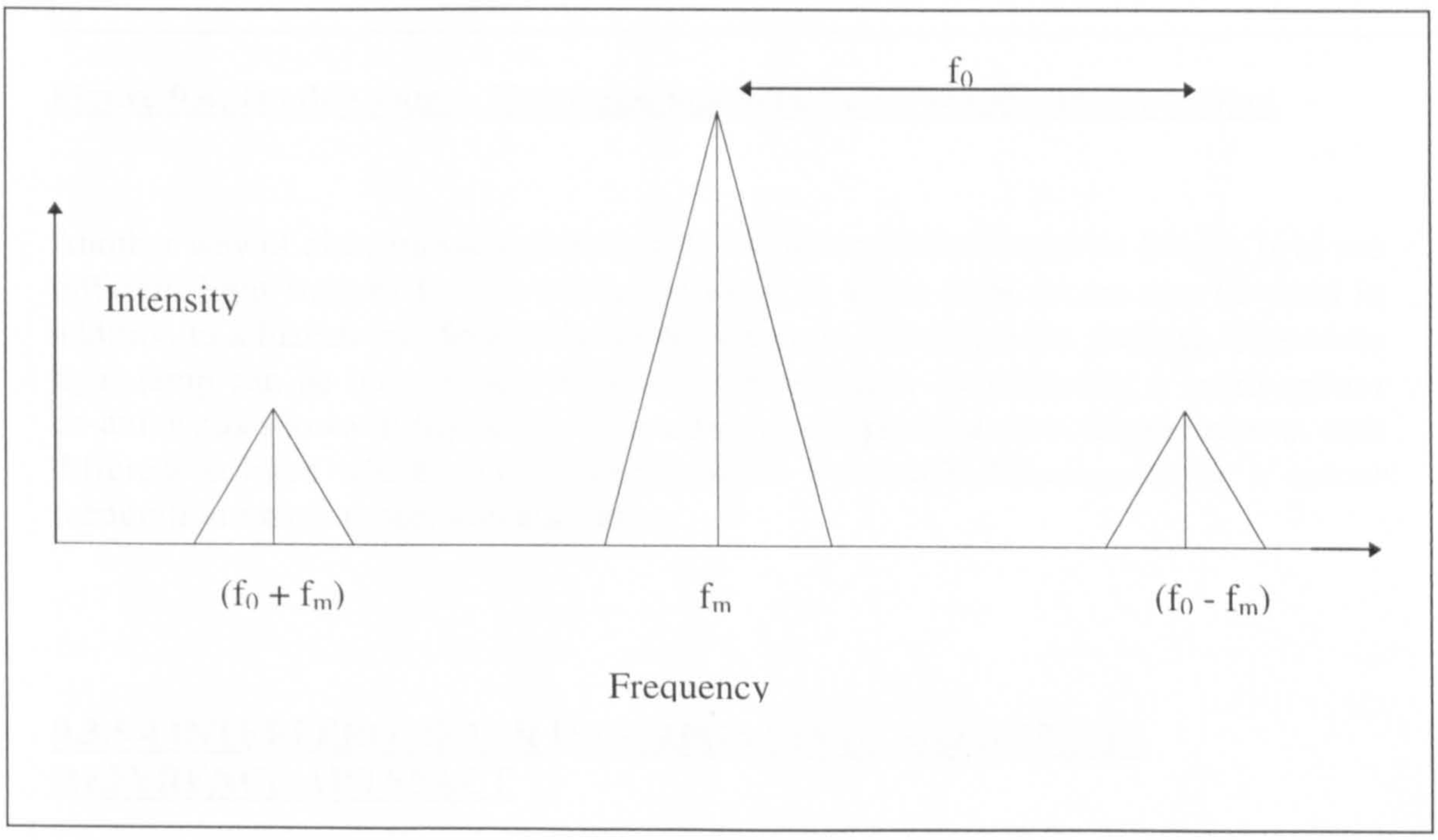


Figure 9.4: Spatial frequencies in the Fourier domain

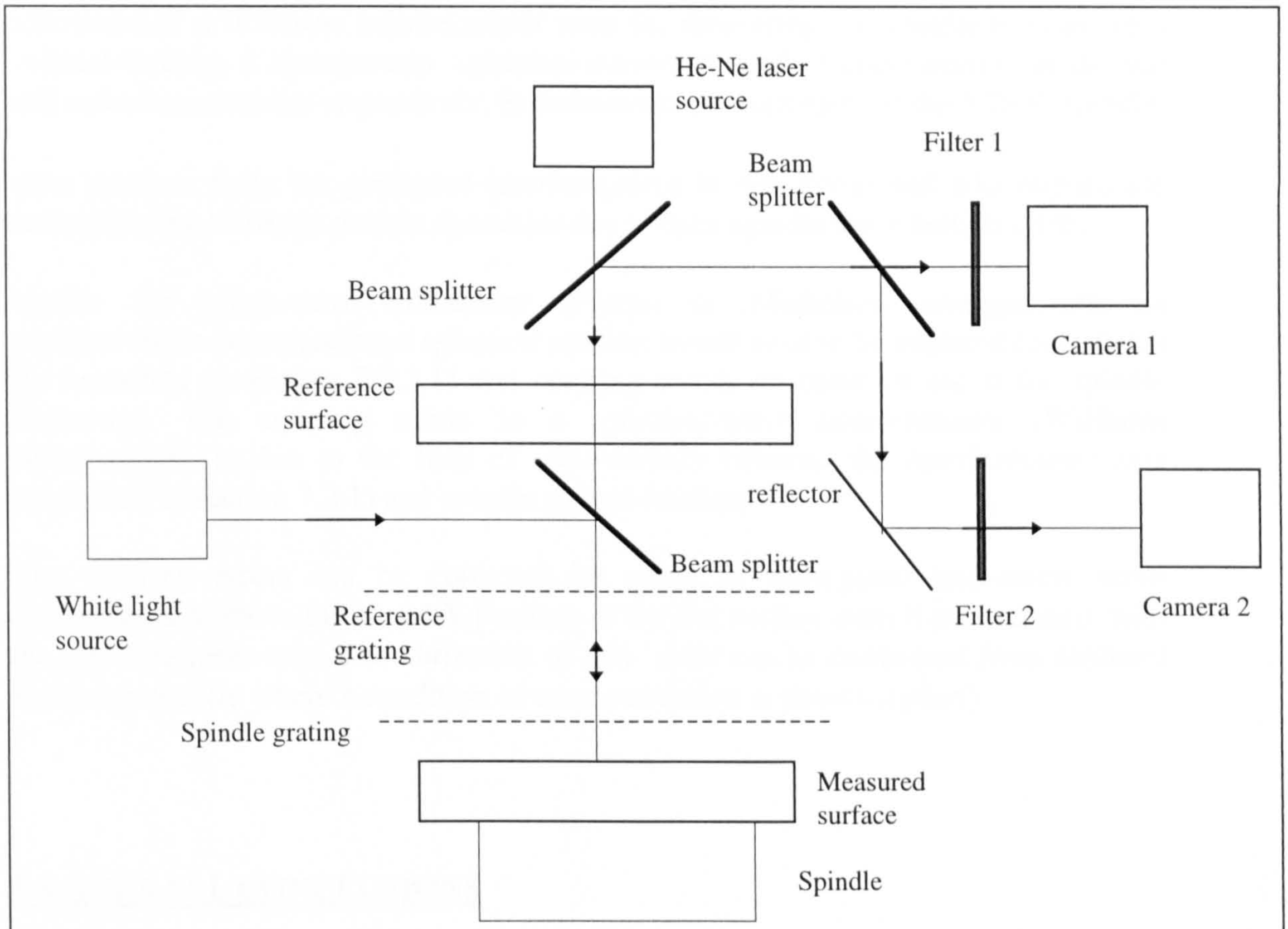


Figure 9.4: Double source Moiré and interferometric spindle measurement

Another way of discriminating between the Moiré and interferometric fringes is to use different light sources for the two techniques. A white light source can be used in addition to a Fizeau interferometer He-Ne source to illuminate the gratings. The white light lamp can be incorporated with the interferometer by mounting a beam-splitter co-axially as shown in figure 9.4. This arrangement produces two fringe patterns with different colours which can be discriminated by simple filtering and/or a colour recognition image processing algorithm.

9.3.5.4 INTERFEROMETER INCORPORATING A *SPHERICAL* REFERENCE ARTEFACT

An interferometric arrangement which incorporates a *spherical* mirror, would be sensitive to spindle radial tilt and axial error motions (all 5 DOF). Examples of such are those due to the Williams interferometer (described in section 4.7.4.2) and the common-path interferometer (described in section 4.7.4.4).

In section 4.7.4.2 it was pointed out that an Williams interferometer can be used to measure all 5 DOF spindle error motions simultaneously. Figure 4.26 shows a

schematic of a Williams interferometer used for measuring the spindle run-out of a vertical bearing, it incorporates spherical mirrors instead of plane mirrors as the *test* and *reference surfaces* respectively. Simultaneous measurement of the 5 DOF spindle

error motions from the generated interferograms is non-trivial and will require the understanding of fringe pattern dynamics due to each spindle error motion DOF.

Unlike the plane-wave instrument (Fizeau or Michelson arrangement), an interferometer incorporating a spherical artefact would need to be adjusted for both tilt (ie described in section 7.2.3.1) and centring errors, in order to use it for spindle metrology. The centring errors in a spherical-wave interferometer (Williams arrangement), is due to the lack of concentricity between the interferometer axis (described in section 7.2.1) and spindle axis of rotation.

The centring errors can be corrected by using an orthogonal lead-screw servo arrangement to move the X and Y position of the *test* surface until it is concentric with the interferometer axis. The correction of this error can be confirmed from captured inteferograms (ie where a condition of error correction is known *a piori*).

9.4 OVERALL CONCLUSIONS

This work has demonstrated the capability of measuring spindle error motions to nanometric accuracy using optical interferometry. It has also demonstrated the suitability of analysing inteferogram data (obtained *on-the-fly*) using intensity-based fringe analysis to obtain values of fringe maximum (to an accuracy greater than one-hundredth of a pixel as described in section 6.5.2), and automatic adjustment of the inteferometer using lead-screw servos (see in section 7.2.3.1) based on dither (see in section 7.7.2) and adaptive model reference (see in section 7.7.3) control regimes.

The measured and estimated spindle error motion data in this work correlate very well with those stated by the bearing manufacturer, with slight deviations between the measured and stated bearing run-out of less-than 10 nm and less-than 0.2 μrads for axial and tilt error motions respectively. Reasons for these deviations have been described in section (8.6.1). In general the results obtained in this work confirm the suitability of using optical interferometry for measuring spindle run-out. It is recommended in section 9.3, that the spindle metrology system described in this thesis could be modified to:

- accommodate high-speed interferogram acquisition/analysis (required for the analysis of high speed spindles as recommended in section 9.3.2 and 9.3.3)

- reduce the influence of environmental variations on refractive index, by using a common-path interferometer or a refractometer (described in section 9.3.4)

- measure all 5 DOF spindle error motions by using a hybrid interferometric and capacitive or (inductive) technique (see in section 9.3.5.2) or a spherical-wave interferometer (described in section 9.3.5.3).

Finally, it is hoped that this work has contributed to the knowledge of spindle metrology and that developments in this work will improve future manufacture, metrology and application of precision machine tools, measuring instruments, rotary electro-mechanical devices and other areas of precision engineering which directly/indirectly use ultra precision spindles.

ACKNOWLEDGEMENTS

I wish to thank my supervisor Dr A E Gee, for without his encouragement, advice and guidance, this thesis would not have been presented in its current form.

I would also like to acknowledge Mr D Simpson (my current senior at NPL) for encouraging me to complete this thesis.

Finally I wish to thank my father and mother, for the general advice and encouragement they have given me over the years towards my education.

REFERENCES

Anderson, J. D. (1995) "Computational fluid dynamics - the basics with applications" McGraw-Hill, New York, NY, pp 1-30

Arneson, H.E (1969) "Externally pressurised bearing structure" US Patent No. 3,472,568

ASME/ANSI (1990) "Methods for performance evaluation of coordinate measuring machines" B89.1.12M

ASME/ANSI (1985) "Axes of rotation standard" B89.3.4

Astrom, K. J. & Wittenmark, B. (1995) "Adaptive control" Second edition, Addison-Wesley, Reading, MA, pp 42 - 56 and pp 185-199

Augustyn, W.A (1973) "Common path interferometry" Optical Engineering, Vol 12, pp180-181

Awcock, G.J. & Thomas, R. (1995) "Applied image processing" McMillan Press, Hampshire, pp 93-99, pp 165-174

Baker, R. E. & Hornung, K . G. (1970) "Effects of heat generation in an air hydrostatic journal bearing" Trans ASME: J. Lub. Tech., Vol 92, pp 607-616

Bar-kana, I. & Kaufman, H. (1984) " Some examples of direct adaptive control to large structural systems" J. Guidance, Vol 7, pp 171-724

Becker, F. & Yu, Y.H. (1985) "Digital fringe reduction techniques applied to the measurement of three-dimensional transonicflow fields" Optical Engineering, Vol 24, pp 429-434

Bentley, J.P, (1988) "Principles of measurement systems" "2nd ed, Longman, Harlow,
pp 155-162

Bryan, J. B., Clouser, R. W, & Holland . E. (1967) "Spindle accuracy" American Machinist, Special Report No 612, pp 149-163

Bryan, J. B. & Vanherck, P. (1975) "Unification of terminology concerning the error motion of axis of rotation" Ann CIRP, Vol 24, pp 555-562

Bryan, J. B., (1968) "International status of thermal error research" Ann CIRP, Vol 16, pp 203-215

BSI (1989) "Coordinate measuring machines" Parts 1-3, BS 6808

Burch, J. M., (1958) "Scatter fringes of equal thichness" Nature, Vol 171, pp889 - 890

Castelli, V. & Pirvics, J. (1968) "Review of numerical methods in gas bearing film analysis" Trans. ASME, J. Lub. Tech., Vol 4, pp 777-792

Chang, M., Lin, P.P. & Tai, C.W. (1994) "A fringe centre detection technique based on a sub-pixel resolution and its applications using sinusodial gratings" Optics and lasers in Engineering, Vol 20, pp 163-176

Chapara, S. C & Canale, R. P, (1990) "Numerical methods for engineers" 2nd Edn, McGraw-Hill, New York, NY, pp 400 - 425

Chapman, P. D. (1985) "A capacitance based ultra-precision spindle error analyser" Precision Engineering, Vol 7, pp 129-137

Chen, M. Q & Yang, C. H. (1989) "Dynamic compensation Technology of spindle error motion of a precision lathe" Precision Engineering, Vol 11, pp 135-138

Collier, P & Wilison, R. (1982) "Engineering Drawing" Hutchinson, London, pp 211 - 220

Cook, P. A (1994) "Nonlinear dynamical systems" Prentice-Hall, Hemel Hempstead, pp 158-169

Creath, K. (1993) "Temporal phase measurement methods" pp 95-112 of Robinson, D.W. & Reid, G. T. (eds) "Interferogram analysis: Digital fringe pattern measurement techniques" IoP, Bristol/Philadelphia, PA

Culshaw, B.& Dakin, J. (1988) "Optical fiber sensors: principles and components" Vol 1, Artech House, Norwood, MA, pp 12-15

Culshaw, B.& Dakin, J. (1989) "Optical fiber sensors: systems and applications" Vol 2, Artech House, Norwood, MA, pp 330-331

Dalsa Inc, (1995) "1996-1997 Databook: Advancing image capture technology", Dalsa, Waterloo, Ontario, Canada, pp261-268

DDC Corp, (1994) "Synchro/resolver conversion handbook",4th edn, DDC, Bohemia, NY, pp125-153

Doung-chul, H., Sang-shin, P., Woo-jung, K. & Jong-woo, K. (1994) "A study on the characteristics of externally pressurized air bearings" Precision Engineering, Vol 16, pp 164-173

Doebelin, E.O. "Measurement systems: application and design" 4th edn, McGraw-Hill, New York, NY

Doo, R., Rodkiewicz, C. M., & Gupta, R. N., (1985) "Development of transient temperatures in a thrust bearing" Trans ASME: J. Tribology, Vol 107, pp 280-283

Duduch, J. G. (1993) "Some critical aspects of machine design and performance for the machining of brittle materials" PhD Thesis, Cranfield, pp 58-63

DiPrama, R. C. (1971) "The equations of motion" in "Design of gas bearings" Mechanical Technology Inc., Lotham, NY, Vol 1, pp 3.1.1 - 3.1.5

Dyson, J. (1970) "Interferometry as a measuring tool" Machinery Publishing, Brighton, pp 129 - 146

Erwin, G.L. (1967) "Interferometers and associated devices" in "Handbook of industrial metrology" American Society of Tool and Manufacturing Engineers, Prentice-Hall, Englewood Cliffs, NJ, pp 246-286

Fawcett, S.G. & Keltie, R.F. (1990) "Use of a fiber optic displacement probe as a surface finish probe" Sensors and Actuators A , Vol 24, pp 5-14.

Franklin, G. F., Powell, J. D. & Emami-Naeini, A. (1994) "Feedback control of dynamic systems" Addison-Wesley, Reading, MA, pp 118 -126

Garcia-Suarez, C., Bogy, D . B & Talke, F. E. (1984) "Use of an upwind finite element scheme for air-bearing calculations" ASLE, SP-12, pp 90-96

Gasvik, K.J. (1995) "Optical metrology" 2nd edition, John Wiley, Chichester

Gee, A. E., McCandlish, S. G. & Puttick, K. E. (1988) "Interferometric monitoring of spindle and workpiece on an ultra-precision single-point diamond facing machine" Proc. SPIE, Vol 1015, pp 74-80

Gee, A. E., Green, D. & Payne, D., (1987) "An in-process tool proximity sensor for ultra-fine machining" Surface Topography, Vol 1, pp 181 - 191

Gee, A. E. (1993) "Dynamic rotational interferometry" pp 203-210 of Blackshaw, D. M. S., Hope, A. D. & Smith, G. T. (eds) "Laser metrology and machine performance" Computational mechanics, Southampton/Boston (Proc. 'LAMDAMAP 93' Conf. Southampton)

Grassam, N. S. & Powell, J. W. (1964) "Gas lubricated bearings" Butterworths, London, pp 263-280

Green, D.E. (1987) "On optical proximity sensor incorporating CCD imaging and signal processing" BSc (Engineering) Dissertion, University of Surrey, pp 35-38

Ginsberg, J. H. (1995) "Advanced Engineering Dynamics" Cambridge University Press, pp 55-73

Gross, A. G.,Matsch, L. A., Castelli, V., Eshel., A., Vohr, H. G. & Widmann, M. (1980) "Fluid film lubrication" John-Wiley, New York, NY, pp550 - 628

Hang, C. & Parks, P. (1973) "Comparative studies of model reference adaptive control systems" IEEE Trans. Automat. Contr, Vol AC-18, pp 419- 28

Hansen, H. (1988) "A new dynamic spindle analyser" Proc ASPE Conf. Atlanta, pp 52-56

Han, D., Park, S., Kim, W. & Kim, J., (1994) "A study on the characteristics of externally pressurised air bearings " Precision Engineering, Vol 16, pp 164-173

Hanes, G. & Dahlstrom, C. E. (1969) "Iodine hyperfine structure observed in saturated absorption at 633 nm" Apllied Physics Lett., Vol 14, pp 362 -364

Hariharan, B.F., Oreb, B.F. & Eiju, T. (1987) "Digital phase-shifting interferometry: a simple error-compensating phase calculation algorithm" Applied Optics, Vol 26, pp 2504-2506

Harrison, W. J. (1913) "The Hydrodynamic theory of lubrication with special reference to air as a lubricant" Trans. Camb. Phil. Soc., vol 22, pp 39-54

Hendricks, F. (1988) "A design tool for steady gas bearings using finite elements, APL language and Delaunary triangulation" STLE, SP-25, pp 124-129

Hirn, G. A. (1864) "Sur Les principaux phenomenes qui presentent les frottements mediats" Bulletins Societe Industrielle de Mulhouse, vol 26, pp 188-277

Horne, D. F. (1983) "Optical Technology" 2nd edn, Adam Hilger, Bristol, pp379

Horikawa, O., Sato, K. & Shimokohbe, A. (1992) "An active air bearing" Nanotechnology, Vol 3, pp 84-90

Hosoe, S. & Tanak, S. (1995) "A low cost method of obtaining a stable environment of a highly precise displacement measuring laser interferometers" Nanotechnology, Vol 6, pp 24-28

Houffman, E., Spur, G., Paluncic, K., Benzinger, H. & Nymoen, H. (1988) "Thermal behaviour optimization of machine tools" Ann CIRP, Vol 37, pp 401-405

Houston, J. B., Buccini, C. J & Neil, P. K. (1967) "A laser unequal path interferometer for the optical workshop" Applied Optics, Vol 6, pp 1237-1242

Hunter, J.C & Collins, M.W. (1990) "The semi-automatic analysis of compressible flow interferograms" Meas. Sci. Technol., Vol 1, pp238-246

Huntley, J.M. (1989) "Noise-immune phase unwrapping algorithm" Applied Optics, Vol 28, pp 3268-3271

Hughes, S. J., Hogg, S. I. & Jones, T. V. (1992) "Analysis of a gas lubricated hydrodynamic thrust bearing" Trans ASME: J. Lub. Tech., Vol 118, pp 449 - 456

Idowu, A. O. & Gee, A. E. (1996) "Wide aperture interferometric spindle metrology: sesitivities and strategies" pp 182-187 of Grattan. K. T. V. (ed) "Applied optics and optoelectronics" (Proc. Appl. Opt. Div. Conf. Reading, 1996)

Ifeachor, E. C. & Jervis, B. W. (1993) "Digital signal processing" Addison-Wesley, Workingham, pp 251 - 276

Iordanoff, I., Sefan, P., & Boudet, R., (1995) "Dynamic analysis of thrust bearing-effect of misalignment and load" Proc. Inst. Mech. Eng., Vol 209, pp 189-194

Isermann, R., Lachmann, K., & Matko, D., (1992) "Adaptive control systems" Prentice-Hall, London, pp 245 - 246

Jedrzejewski, J., & Madrzycki, W., (1992) "A new approach to modelling thermal behavior of a machine tool under service conditions" Ann CIRP, Vol 41, pp 455-458

Jenkins, F.A, & White, H.E. (1976) "Fundamentals of optics" 4th edition, McGraw-Hill, New York city, pp 232-259

Jones, R. V (1956) "Some parasitic deflexions in parallel spring movements" J. Sci. Instrum., Vol 33, pp 11- 15

Kato, T. & Obara, S. (1996) "Improvement in dynamic characteristics of circular journal bearings by means of longitudinal microgrooves" Tribology Trans., Vol 39, pp 462-468

Kilt, P. & Lund, J. W. (1986) "Calculation of dynamic coefficients of a journal bearing using a variational approach" Trans. ASME, J. Tribology, Vol 108, pp 421-425

Kim, S. & Park, Y. (1994) "Optical measurement of spindle radial motion by moiré technique of concentric-circle gratings" Int. J. Mach. Tools Manuf., Vol 34, pp 1019-1030

Kingsbury, A. (1877) "Experiments with an air lubricated journal" J. Am. Soc Nav. Eng., Vol 9, pp 267-292

Kumada, Y., Hashizume, K. & Kimura, Y. (1996) "Performance of plain bearings with circumferential microgrooves" Tribology Trans., Vol 39, pp 81-86

Kujawinska, M. (1993) "Spatial phase measurement methods" pp 141-193 of Robinson, D.W. & Reid, G. T. (eds) "Interferogram analysis: Digital fringe pattern measurement techniques" IoP, Bristol/Philadelphia

Lion Precison (1992) "SEA 2.30 spindle error analyser" St Paul, MN,Report No NC801-194.5, pp 1 -12

Low, A. (1991) "Introductory computer vision and image processing" McGraw-Hill, Maidenhead, pp165-189

Malanoski, S. B. & Pan, C. H. T. (1965) "The static and dynamic characteristics of spiral grooved thrust bearing" J. Basic Eng., Trans. ASME, Vol 87, pp 547-558

Malanoski, S. B & Pan, C. H. T. (1971) "Fixed thrust bearings" in "Design of gas bearings" Mechanical Technology Inc., Lotham, NY, Vol 2, pp 6.3.1 - 6.3.2

Marion, A. (1991) "An introduction to image processing" Chapman & Hall, London, pp 35-49

Marshall, S.J., Caulfied, M.M. & MacKenzie, P.M. (1986) "The application of automatic fringe analysis in fracture mechanics" Optics and lasers in engineering, pp 175-193

Martin, D. (1996) "Using capacitance to measure the limit of machine contouring performance" Proc ASPE Conf. Austin, pp 43-47

McKeown, P. A. (1986) "High precision manufacturing" Proc Inst. Mech. Eng, Vol 200, pp 1-19

McKeown, P., Weck, M., Bonse, R., & Herbst, U. (1995) "Reduction and compensation of thermal errors in machine tools" Ann CIRP, Vol44, pp 589-598

Mizuno, H. (1993) "Micro-tilt controlled rotating face-plate for single-point diamond turning" PhD Thesis, Cranfield, pp 17-22, 98-99

Mizumoto, H., Matsubara, N., & Usuki, M. (1990) "Zero-compliance aerostatic bearing for an ultra-precision machine" Precision Engineering, Vol 12, pp 75-80

Nica, A. (1970) "Thermal behavior and friction in journal bearings" Trans ASME J. Lub. Tech. Vol 92, pp 373-380

Ohsumi, T., Mori, H. & Ikeuchi, K., (1984) "A study on the vibratory energy dissipation in an externally pressurised lubricating film" J. JSLE Int. Ed, Vol 5, pp 89-94

Orcutt, F.K. (1971) "Instrumentation and measurement" in "Design of gas bearings" Mechanical Technology Inc., Lotham, NY, Vol 2, pp 10.1.1 - 10.1.11

Osburn, P. V. & Whitaker, H. P. (1961) "New developments in the design of adaptive control systems" Paper No. 61-39, Institute of Aeronautical Sciences, MIT, pp 1-15

Otsu, N. (1979) "A threshold selection from grey-level histograms" IEEE Trans: Sys., Man, Cybern.,Vol SMC-9, pp 62-66

Pan, C. H. T. (1990) "Gas lubrication (1915-1990)" pp 31-55 of ASME/STLE Tribology conference "Achievements in tribology 90" ASME, New York, NY

Parker, J.R.(1994) "Practical Computer vision using C" John Wiley, Somerset, NJ, pp 9-12, pp 126-145

Pederotti, F.L. & Pederotti, S.L. (1993) "Introduction to optics" 2nd edition, Prentice-Hall, New York, NY

Phillips, C. H. & Nagle, H. T. (1990) "Digital control system analysis and design" Prentice-Hall, Englewood Cliffs, NJ, pp 377 - 386

Powell, J. W. (1970) "Design of aerostatic bearings" Machinery Publishing, Brighton
,
pp 15-39

Pugh, D.J. & Jackson, K. (1982) "Automatic guage block measurement" Proc NELEX' 82 Metrology Conference NEL, East Kilbride, Scotland, pp 1-15

Puttick, K. E., Rudman, M. R., Franks, A. & Lindsey, K. (1989) "Single-point diamond machining of glass" Proc. R. Soc. London, Vol A426, pp 19-30

Ramesh, V. (1990) "Development of a two-axis optically sensed scanning contact probe for use with coordinate measuring machine" MSc Thesis, Cranfield, pp 2-4.

Reynolds, O. (1886) "On the theory of lubrication and its application to Mr Beauchamp Tower's experiments, including an experimental determination of viscosity of olive oil" Phil. Trans. R. Soc. London, Ser A, Vol 177, pp 157-234

Reid, G.T (1986) "Automatic fringe pattern analysis: a review" Optics and Lasers in Engineering, Vol 7, pp37-68

Risse, S. & Guyenot, V. (1996) "Aerostatic precision glass bearing with an integrated polygonal mirror for scanning - a new method" Proc. SPIE, Vol 2775, pp 619-628

Reiger, N. F., (1971) "Bearing rotor dynamics" in "Design of gas bearings" Mechanical Technology Inc., Lotham, NY, Vol 2, pp 8.1.1 - 8.1.40

Robinson, D.W. & Williams, D.C. (1986) "Digital phase stepping speckle interferometry" Optics Communications, Vol 57, pp 26-30

Robinson, D.W. & Reid, G.T. (1993) "Interferogram analysis - digital fringe pattern measurement techniques", IOP Press, Bristol, pp 1 -21

Russ, J.C. (1995) "The image processing Handbook" CRC Press, Boca Raton, FL, pp 290-308

Sawtell, W. H. (1909) "Some points regarding transverse spindle grinders" American Machinist, Vol 32, pp 248-249

Scarr, A.J. (1967) "Metrology and precision engineering" McGraw-Hill, London, pp 72-76

Schellekens, P. (1986) "Design and results of a new interference refractometer based on a commercially available laser interferometer" Ann CIRP, Vol 35 pp 387-397

Schlesinger, G. (1938) "Testing machine tools" Machinery Publishing, Brighton, pp 29-30

Schwider, J., Burow, R., Elssner, K.E., Grzanna, J., Spolaczyk, R. and Merkel, K. (1983) "Digital wavefront measuring interferometry: some systematic error sources" Applied Optics, Vol 22, pp 3421-3432.

Shinners, S. M. (1979) "Mordern control system theory and application" Second edition, Addison-Weseley, Reading, pp 369-401

Shrinivasa, N., Ziegert, J. C., & Mize, C. D., (1996) "Spindle thermal drift measurement using the laser ball bar" Precision Engineering, Vol 18, pp 118-128

Sinha, N. K. (1986) "Control systems" CBS, New York, NY, pp 13 - 14, pp 259 - 279

Sinha, N. K. & Kuszta, B. (1983) "Modeling and indentification of dynamic systems" Van Nostrand Reinhold, New York, NY, pp 51 - 75

Slocum, A. H. (1992) "Precision machine design" Prentice-Hall, New York , NY

Slocum, A. H., Annaswamy, A.M & Smith, M. H. (1995) "Adaptive control strategies for a presicion machine tool axies" Precision Engineering Vol 17, pp 192-206

Sonka, M., Hlavac, V. & Boyle, R. (1993) "Image processing, anlysis and machine vision" Chapman & Hall, Cambridge, pp 112-121, pp 422-439

Smith, S.T. & Chetwynd, D.G (1992) "Foundations of ultra-precision mechanical design" Gordon and Breach, London, pp 207

Sneck, H. (1991) "Machine Dynamics" Prentice-Hall, Englewood Cliffs, NJ, pp 192-206

Stadler, W. (1995) "Analytical robotics and mechatronics" McGraw-Hill, New York, NY, pp 54 - 76

Streeter, V. L. (1961) "Handbook of fluid dynamics" McGraw-Hill, London, pp 1-19

Stone, W., (1921) "A proposed method for solving some problems in lubrication" Commonwealth Engr, Vol 6, pp 114-149 (to be checked !!)

Surrel, Y. (1993) "Phase stepping: a new self-calibrating algorithm" Applied Optics Vol 32, pp 3598-3600

Takeda, M., Ina, H. & Kobayashi, S. (1982) "Fourier-transform method of fringe-pattern analysis for computer-based topography and interferometry" J. Opt. Soc. Am., Vol 72, pp156-160

Talke, F. E. (1987) "Precision engineering issues of Magnetic recording technology" Precision Engineering, Vol 9, pp 171 - 178

Tower, B. (1885) "Second report on friction experiments" Proc. Inst. Mech. Eng., Vol 36, pp 58-70

Tlusty, J. (1959) "Systems and methods of testing machine tools" Microtechnic, Vol 13, pp 162-178

Triandis, I., Mizuno, H., Hara, Y. & Grafandis, D. (1991) "An interferometrically monitored and controlled micro stage" MSc Metrology & Quality Assurance Group Project Report, School of Industrial and Manufacturing Science, Cranfield University, pp 22-32

Venugopal, R., & Barash, M., (1986) "Thermal effects on the accuracy of numerically controlled machine tools" Ann CIRP, Vol 35, pp 255-258

Visilog (1993) "A tutorial on image processing" Manchester Computing Centre, University of Manchester, pp 1-184

White, J. W. & Nigam, A. (1980) "A factored implicit scheme for the numerical solution of the Reynolds equation at very low spacing" Trans ASME, J. Lub. Tech, Vol 102, pp 80-85

Wilkie, D & Fisher, S. A, (1963) "Measurement of temperature by Mach-zehnder interferometry" Proc. Inst. Mech. Eng., Vol 178, pp 461-470

Williams, J. A. (1994) "Engineering Tribology" Oxford University Press, pp 301-320

Williams, D.C., Virdee, M.S., Nassar, N.S. & Banyard, J.E. (1991) "Digital phase-step interferometry: a simplified approach" Optics and Laser Technology, Vol 23, No 3, pp 147-150

Williams, D. C (1993) "Optical methods in engineering metrology" Chapman & Hall, London, pp 159 - 162

Wilson, J. & Hawkes, J.F. (1989) "Optoelectronics: an introduction" 2nd edn, Prentice-Hall, New York, NY

Wunsch, H. L. (1965) "Applications of gas bearings to precision grinding" Gas Bearing Symposium, University of Southampton, Paper no. 9a

Yabe, H. (1994a) "A study on run-out characteristics of externally pressurised gas journal bearing (rotor run-out characteristics)" JSME Int. J., Ser C, vol 37, pp 355-361

Yabe, H. (1994b) "A study on run-out characteristics of externally pressurised gas journal bearing (modified DF method for point-source solution)" JSME Int. J., Ser C, vol 37, pp 362-368

Yabe, H., & Ishida, H. (1991) "A study on the running accuracy of an externally pressurized gas thrust bearing (rotor run-out characteristics)" JSME Int. J., Vol 34, pp 333-338

Yao-Sun, M.S. (1994) "Axis of rotation analyser" Ann CIRP Vol 42, pp 439-444

Yatagai, T. (1991) "Automated fringe analysis techniques in Japan" Optics and lasers in engineering, Vol 15, pp79-91

Yatagai, T. (1993) "Intensity based analysis methods" pp 73-94 of Robinson, D.W. & Reid, G. T. (eds) "Interferogram analysis: Digital fringe pattern measurement techniques" IoP, Bristol/Philadelphia, PA

Yoon-Chang, P. & Seung-woo, K. (1994) "Optical measurement of spindle radial motion by Moire technique of concentric-circle gratings" Int. J. Mach. Tools Manufact., Vol 34, pp 1019-1030

APPENDIX 1: ESTIMATION OF BEARING DAMPING COEFFICIENT

According to Powell (1970), the bearing damping coefficients can be estimated using the equation:

$$c_{d_z} = \frac{LBp_r W_T}{2rN} \tag{1}$$

where,
p_r: approximate pressure in spindle journal and thrust bearing
W_T: total load acting on spindle journal and thrust bearing
P_s = 1034 kPa (bearing air supply pressure)
B = 0.1 m (width of bearing rotor)
L = 0.1 m (length of bearing rotor)
N = 10,000 rev/min (maximum speed of bearing)
r = (L/2) = 0.05 m (radius of bearing rotor)

The evaluation of the p_r was given in section 2.5.4 as:

$$p_r = p_s\left(1-\left(\frac{\alpha-1}{2}\right)\left(\frac{v}{a_o}\right)^2\right)^{\left(\frac{\alpha}{\alpha-1}\right)} \tag{2}$$

and if velocity of sound in the bearing is $a_o = \sqrt{\alpha TR} = 312.957$ m/s (3)

and velocity of air flow in the bearing $v = \frac{\dot{m}}{hd\pi} = 604.789$ m/s (4)

Hence p_r = 886.2 kPa

The value of W_T can be calculated using the equation:

$$\frac{p_r}{LB} = 10.03 \text{ kN} \tag{4}$$

By substituting p_r and W_T into equation (1), the values of spindle coefficients are:

$$c_{d_z} = 9.163 \times 10^{10} \text{ N/ms}^{-1} \tag{5}$$

$$c_{d_r} = c_{d_z}\left(\frac{f_{d_r}}{f_{d_z}}\right) = 4.581 \times 10^{10} \text{ N/ms}^{-1} \tag{6}$$

$$c_{d_t} = c_{d_r} = 4.581 \times 10^{10} \text{ N/ms}^{-1} \tag{7}$$

CRANFIELD UNIVERSITY

SCHOOL OF INDUSTRIAL MANUFACTURING SCIENCE

Ph.D. THESIS - PROGRAM LISTINGS

Academic Year 1997-98

ADE IDOWU

DYNAMIC METROLOGY OF ERROR MOTIONS IN PRECISION SPINDLES USING OPTICAL METROLOGY

Supervisor: Dr A E Gee

October 1998

INTRODUCTION

This document contains a list of all the software programs developed for this project, these include

PROGRAM LISTINGS 1: ALGORITHMS USED TO SAMPLE INTERFEROGRAM DATA

```
/************* FILENAME: C:\VISION_P\MATLAB_RSC\DATA_FRINGE1.RSC ****************/

  /**
      PROGRAM DESCRIPTION: This program is used to calculate fringe data at specified
      angular positions:

      fringe_density
      spatial frequency
      fringe inclination
      fringe maxima

DEVELOPED BY: ADE IDOWU  (2/2/98)
Version 4.1

                        **/

#include "c:/vision_p/maxima_1.rsc"
#include "c:/vision_p/tre_new2.rsc"
#include "c:/vision_p/minfinda.rsc"
#include "c:/vision_p/store_4a.rsc"
#include "c:/vision_p/stored_5.rsc"
#include "c:/vision_p/stored_9.rsc"
#include "c:/vision_p/scaled_1.rsc"
#include "c:/vision_p/phase2.rsc"
#include "c:/vision_p/coords.rsc"

 test()
 {

// Definition of program variables

   int g, h1[300], h2[300], k[300], j,  i,num,z, no_of_pixels;
   float l, count[300],m;
   int q, loop_1,loop_2,fringe_count, x_loc, y_loc[300], intensity[300];
   char file[100],result[200], a[300],b[300], c[300],d[300],e[300],f[300];
   char a_1[300], file_1[100],a_2[300], file_2[100],a_3[300], file_3[100];;

 loop_1 = 1;
 loop_2 = 30;

// Definition of read/write files

 for (i=1; i <=loop_1 ; i++)  {
          a[i] = "c:/vision_p/data_new/metro_";
```

```
        a_1[i] ="c:/vision_p/mat_lab/data_";
        a_2[i] ="c:/vision_p/mat_lab/info_"
        a_3[i] ="c:/vision_p/mat_lab/fringe_"
        b[i] = i;
        c[i] =".txt";


strncat(a[i], b[i], 10000000);
strncat(a[i], c[i], 10000000);
file[i] = a[i];

strncat(a_1[i], b[i], 10000000);
strncat(a_1[i], c[i], 10000000);
file_1[i] = a_1[i];

strncat(a_2[i], b[i], 10000000);
strncat(a_2[i], c[i], 10000000);
file_2[i] = a_2[i];


strncat(a_3[i], b[i], 10000000);
strncat(a_3[i], c[i], 10000000);
file_3[i] = a_3[i];


 for (j=1; j <=loop_2 ; j++)  {
    d[j] = "c:/vision_p/metro/m1_";
    e[j] =  j*(360/loop_2);
    f[j] = i;
strncat(d[j], e[j], 10000000);
strncat(d[j], f[j], 10000000);
result[j] = d[j];

count[j] = j;

//Output data are:

// g : fringe density
// h : fringe position
// k : fringe intensity
// l : fringe orientation
// m : spatial frequency


  mini_bin(result[j], &g, &h1,&h2, &k,&l, &m);

  printf("Spatial frequency: %lf\n", m);
  printf("Orientation: %lf\n", l);

// Store fringe density, spatial frequency, fringe oirentation,

  storage4(file[i],j,g,l,m);
  if (g>1) {
```

```
// Store fringe maxima data
   storage9(file_3[i],g);
    for (q = 1; q<=g; q++)
 {

// Store fringe pixel data
    storage5(file[i],h1[q],h2[q]);
    if (h2[q] == 0)
        {
                fringe_count = (g - 1);
        }
if (h2[q] > 0)
{
        printf("h[q] is positive\n");
  no_of_pixels = h2[q] - h1[q];
        // Store fringe pixel data
        storage5(file_2[i],h1[q],h2[q]);
        for (z = h1[q]; z<=h2[q]; z++)
        {
                y_loc[z] = z;
                x_loc = 128;
                IpReadPix("new_rot2", x_loc, y_loc[z] , &intensity[z]);
                printf("Data for fringe location %d is %d\n",y_loc[z], intensity[z]);
                // Store fringe pixel data to be read by MATLAB
                storage5(file_1[i],y_loc[z],intensity[z]);
        }
 }

}
                }

        }

  printf(" \n");
  printf("Data for sample: %d\n",i);
  printf(" \n");
    printf(" \n");
    printf(" \n");
    }

}
```

// Function used acquire fringe data from an interferogram

```
mini_bin(file1, a, b1,b2, c, d, e)

{
```

```
// Definition of program variables

double angle_degrees,rotation_angle;
int high_threshold, low_threshold, level[2],cut_1[2];
char filename;
int value, start_y[300],end_y[300], j, in[300];
int x, y_lenght, flag1,temp;
float spatial_frequency;

cut_1[0] = 58;
cut_1[1] = 90;

// Compute interferogram spatial frequency and orientation

        angle_degrees = locating_pixels(file1, &flag1, &spatial_frequency);
        e = spatial_frequency;
        // if there is an infinite fringe spacing

        if (flag1 <= 1)
        {
        printf("   ");
        a = 1;
        b1[1] =0;
        c[1]= 0;
        d = 45;
                }

        // if there is an finite fringe spacing but the fringe orientaion is 90 degress
        if (flag1==2) {
        if (angle_degrees==90)
        {
        d = 90;
        rotation_angle = 180;
        }

        // if there is an finite fringe spacing but the fringe orientaion is 0 degress
        if (angle_degrees== 0)  rotation_angle=d = 90.01;
        d = angle_degrees;
        // If the angle is positive, and not 0 or 90
if (d > 0 && d != 0 && d !=90)
        rotation_angle = 90 - (angle_degrees);
        if (d < 0)
        {
        // If the angle is negative
d = 180 + d;
rotation_angle = 90 - (angle_degrees);
}


        // define filename
        filename = "drain_cut";

        // Evaluate the optimal threshold values of the interferogram
optimal_threshold(filename, &low_threshold, &high_threshold);
```

```
level[0] = low_threshold;
    level[1] = high_threshold;
    Threshold(filename, level, "crab2");
IpSetAoiMode(1);
copy("crab2", "crab2_old");
addmask("crab2_old", "c:/vision_p/images/mask1");
copy("crab2_old", "crab2a");
IpSetAoiMode(0);

    // Rotate interferogram and select an area of interest
    rotation("crab2a", "crab3a", rotation_angle, {26,45}, 0);
    cut("crab3a",{9,25}, {40,40},"crab3b");
Ldilate("crab3b",0,0, "crab4"  );
    scaling("input",rotation_angle);

    // Apply morphological function to imnterferogram
    Ldilate("new3",0,20, "new3a"  );
    Threshold("new3", level, "new4");
    rotation("new4", "new_rot1", rotation_angle, {128,128}, 0);
    rotation("new3a", "new_rot2", rotation_angle, {128,128}, 0);

    // Read fringe pixel intensity at x and y locations
x=128; y_lenght=255;
mini_find("new_rot1", x,y_lenght,&value, &start_y,&end_y, &in);

    // Read fringe maxima data
    printf("The detected fringe minima peaks are: %d\n\n", value);
a = value;
    if (value >=1 && flag1==2) {
    for(j=1; j<=value; j++)
    {
printf("Peak is at position %d\t at intensity: %d\n\n", start_y[j], in[j]);
temp  = start_y[j]; b1[j] =temp;
    temp  = end_y[j]; b2[j] =temp;
temp= in[j]; c[j] = temp;
                                        }
                                        }
                                        }

}
```

```
/****** FILENAME: C:\VISION_P\minima_1.RSC ********/

 /**

      PROGRAM DESCRIPTION: This program is used to detect fringe minima
      DEVELOPED BY: ADE IDOWU  (2/9/97)
      Version 2.1
                                                                **/

float  locating_pixels(file1, flag, spatial_frequency)

 {

    int deno, num, cut_1[2];
    int i, num_of_fringes,lenght_1,x_coord, y_distance, total_error,total_fringe;
    int number_of_maxi_1, number_of_maxi_2,number_of_maxi_3,
number_of_maxi_4,number_of_maxi_5, number_of_maxi_6,number_of_maxi_7;
    int
fringe_error_1,fringe_error_2,fringe_error_3,fringe_error_4,fringe_error_5,fringe_error_6,fringe_error_7;
    int average_fringe_number , average_fringe_error_total, number_1, number_2,number_3,
number_4,peak_number;
    float level[2],th_value,scale, constant,fringe_maxima, temp;
    float denomenator,numerator, angle_in_radians, angle_in_degrees, divisor;
    float square, spacing_1, spacing_2, rotation_angle, scan_lenght;
    float thr[2], maximum,minimum, mean,standard_dev;
    float actual_fringe_maxima;
    long inorigin[2];
    long outorigin[2];
    long size[2];


    MOMENTS[0]=0; MOMENTS[1]=0;
    MOMENTS[2]=0; MOMENTS[3]=0;
    MOMENTS[4]=0; ORIENTATION=0;
    FIRST_POINT_X = 0;
    FIRST_POINT_Y = 0;
    NUMBER = 0;


     init.IWINDOW=0;
     PopUpImageWindow();
     read(file1,"input",3);


        // Define AOI for image processing
```

```
    cut_1[0] = 58;
    cut_1[1] = 90;
cut("input", {60,80}, {128,128}, "output");
normalize("output",0, {0,255}, {0,160}, "drain");
subtract("drain", "drain", "tem");
cut("drain", {50,15}, cut_1, "drain_cut");
paste("drain_cut", "tem",{38,19});

    // Apply FFT

 fft2d(
/* Input image */      "tem"
/* Output real part */ ,"real"
/* Output imaginary part */    ,"immag"
);
xytortheta(
/* Input real part */  "real"
/* Input imaginary part */     ,"immag"
/* Output modulus */   ,"mod"
/* Output phase */     ,"phase"
);

center(
/* Input image */      "mod"
/* Output image */     ,"modul"
);


    // Apply low-pass convulution spatial filter
 lowpass_3x3(
/* Input image */      "tem"
/*   */, {
  1
,   1
,   1
,   1
,   1
,   1
,   1
,   1
,   1
}
/* Automatic scaling */,1
/* Output image */     ,"tem1"
);


    // Evaluate image statistics
extrema("modul");
mean = MEAN;
maximum = MAXIMUM;
minimum = MINIMUM;

    // Convert image to floating-point format
```

```
    convert("modul", 0, "mod_conv");
    lowpass_3x3(
    /* Input image */      "mod_conv"
    /*   */ , {
      1
,     1
,     1
,     1
,     1
,     1
,     1
,     1
,     1
,     1
    }
    /* Automatic scaling */ ,1
    /* Output image */     ,"mod_smo"
    );

        // Enhance intensity of FFT peaks
    multiply("mod_smo", 2, "mod_mult");
    mgradient("mod_mult", "mod_edge");
    subtract("mod_mult", "mod_edge", "mod_grey");

    UpdateLut(NORMAL);
    UpdateLut(NORMALF);



/*********** FIRST FFT PEAK DETECTION ***************************/

 thr[0] = 5;       // high spatial frequency (5,200)
 thr[1] = 200;   // low spatial frequency (200, 255);


Hysteresis(
/* Input image */      "mod_smo"
  , thr
 /* constant threshold value */
  /* varying threshold value */

/* Output binary image */      ,"mod_thre"
/* Number of iterations */     ,1
);


     centroid(
/* Input image */      "mod_thre"
/* Output image */     ,"centroid"
);


     label("centroid", "lab");
```

```
IpNumbe("centroid", "lab", &NUMBER);

FIRST_POINT_X = CRmAllocTab(NUMBER);
FIRST_POINT_Y = CRmAllocTab(NUMBER);

IpLocatn("centroid",0, &FIRST_POINT_X, &FIRST_POINT_Y);

number_1 = NUMBER;
```

```
/********** SECOND FFT PEAK DETECTION ***************************/
    if (number_1  == 3)
    {
    peak_number = number_1;
    flag = 2;
    }

    if (number_1 != 3)   {
thr[0] = 200;        // high spatial frequency (5,200)
thr[1] = 255;   // low spatial frequency (200,255);


Hysteresis(
/* Input image */       "mod_smo"
, thr
/* constant threshold value */
/* varying threshold value */

/* Output binary image */      ,"mod_thre"
/* Number of iterations */     ,1
);


centroid(
/* Input image */       "mod_thre"
/* Output image */      ,"centroid"
);
```

```
    label("centroid", "lab");

    IpNumbe("centroid", "lab", &NUMBER);

FIRST_POINT_X = CRmAllocTab(NUMBER);
FIRST_POINT_Y = CRmAllocTab(NUMBER);

IpLocatn("centroid",0, &FIRST_POINT_X, &FIRST_POINT_Y);

number_2 = NUMBER;
```

```
/********** THIRD FFT PEAK DETECTION **************************/

        if (number_2  == 3)
        {
        peak_number = number_2;
        flag =2;
        }


        if (number_2 != 3)
        {
         thr[0] = 5;        // high spatial frequency (5,200)
         thr[1] = 200;   // low spatial frequency (200, 255);

        Hysteresis(
        /* Input image */      "mod_mult"
        , thr
        /* constant threshold value */
        /* varying threshold value */
        /* Output binary image */      ,"mod_thre"
        /* Number of iterations */     ,1
        );


        centroid(
        /* Input image */      "mod_thre"
        /* Output image */     ,"centroid"
        );

        label("centroid", "lab");

        IpNumbe("centroid", "lab", &NUMBER);

        FIRST_POINT_X = CRmAllocTab(NUMBER);
        FIRST_POINT_Y = CRmAllocTab(NUMBER);

        IpLocatn("centroid",0, &FIRST_POINT_X, &FIRST_POINT_Y);

        number_3 = NUMBER;



/********** FOURTH FFT PEAK DETECTION **************************/
        if (number_3  == 3 )
        {
        peak_number = number_3;
        flag = 2;
        }

        if (number_3 != 3)
        {
```

```
thr[0] = 5;        // high spatial frequency (5,200)
thr[1] = 200;   // low spatial frequency (200,255);


     Hysteresis(
/* Input image */      "mod_grey"
, thr
/* constant threshold value */
/* varying threshold value */
/* Output binary image */      ,"mod_thre"
/* Number of iterations */     ,1
);
centroid(
/* Input image */      "mod_thre"
/* Output image */     ,"centroid"
     );

     label("centroid", "lab");
IpNumbe("centroid", "lab", &NUMBER);

     FIRST_POINT_X = CRmAllocTab(NUMBER);
     FIRST_POINT_Y = CRmAllocTab(NUMBER);

     IpLocatn("centroid",0, &FIRST_POINT_X, &FIRST_POINT_Y);

     number_4 = NUMBER;

     if (number_4  == 3 )
     {
     peak_number = number_4;
flag = 2;
}

                                   }
                                     }
                                       }


if(number_1 == 1 && number_2 == 1 && number_3 != 3 && number_4 !=3)
{
angle_in_degrees = 45;
flag = 1; printf("flag: %d\n",flag);  }


if (flag ==2) {

if (peak_number >1 && peak_number <= 10) {
        for(i=0; i<peak_number; i++) {
                                                  }

        if (FIRST_POINT_Y[1] == FIRST_POINT_Y[2] && FIRST_POINT_X[0] >
FIRST_POINT_X[1])
        {
        FIRST_POINT_Y[0] = FIRST_POINT_Y[1];
        numerator   = (FIRST_POINT_Y[0] - FIRST_POINT_Y[2]);
```

```
denomenator = (FIRST_POINT_X[1] - FIRST_POINT_X[0]);
}

else

{
numerator   = (FIRST_POINT_Y[0] - FIRST_POINT_Y[2]);
denomenator = (FIRST_POINT_X[2] - FIRST_POINT_X[0]);
}


// Evaluate the spatial frequency and inclination of the fringe density
numerator = numerator * 100 ; denomenator = denomenator *100;

    if ((denomenator) == 0)     angle_in_radians = 1.570796327;

    if ( (numerator) == 0)   angle_in_radians = 0;

    if (abs(numerator) > 0 && abs(denomenator) > 0)
    {
divisor = division(numerator, denomenator);
    angle_in_radians = atan(divisor);
    }

    angle_in_degrees = angle_in_radians *(division(180, 3.1415927));
    square =( (pow(numerator, 2)) + (pow(denomenator, 2)) );
    spacing_1 = sqrt(square);
    spatial_frequency = spacing_2 = division(spacing_1, 2);
}

        // Alternavtive technique for evaluating firnge spatial frequency and inclination
        IpInertia("centroid", &MOMENTS, &ORIENTATION);


// Dispaly the program outputs (disabled)
//  printf(" The value of the divisor is %f\n", divisor);
//  printf(" The value of 'delta' Y is %f\n", numerator);
//  printf(" The value of 'delta' X is %f\n", denomenator);
//  printf("The orientation of fringe pattern in radians is : %f\n", angle_in_radians);
//  printf("The orientation of fringe pattern in degrees is : %f\n", angle_in_degrees);
//  printf("The fringe spatial frequency is : %f\n\n",spacing_2 );
//  printf("The orientation measured using an alternative approach is : %g deg\n",
ORIENTATION*180/3.1415927);

}

    // Return fringe inclination
    return angle_in_degrees;
}
```

```
/* Function used for carrying out the division of two floating point numbers                    */
/**********START ROUTINE**************************************************************/
 double division(e, f)
{

  int g,h;
 float x;

g = e/f;

x = e - (g*f);

h = x*10000/f;

h = h*0.0001;


x = h+g;

return x;

}
/****************END ROUTINE*************************************************************/
```

```
// FILENAME: C:\VISION_P\MATLAB_RSC\TRE_NEW.RSC

/**

    PROGRAM DESCRIPTION: This program is used to compute
    the optimal treshold of a the fringe pattern based on its
    histogram data.

    DEVELOPED BY: ADE IDOWU  (27/8/96)
    Version 1.5

                **/


optimal_threshold(filename, low_threshold, high_threshold)

{


// Program variables

int min, max, i;
long float sum_pixels, his_togram[500], class_means[500], variance;
long float  cumulative1, cumulative2, cum_pi[500];
long float pi[500], w[500], A, B,  cum_w[500], C, D, E, F, G;


// Compute interferogram satistical data

        extrema(filename);
        min =  MINIMUM;
        max = MAXIMUM;
        histogram(filename, {min, max});
        sum_pixels = HISTOGRAM[0];
        for (i=1; i<=max-min; i++) {
        sum_pixels = sum_pixels + HISTOGRAM[i];
        // printf("The grey-level %d\t has the following number of pixels: %d\n", i+min,
his_togram[i+min]);
      }

        // printf("%d\n", sum_pixels);

cumulative1 = 0;
cumulative2 = 0;

for (i=0; i<=max-min; i++)
his_togram[i+min]  = HISTOGRAM[i];

for (i=0; i<=max-min-1; i++) {
    // calculate occurence of group
```

```
pi[i+min] = division(his_togram[i+min], sum_pixels);
cumulative1 = cumulative1 + pi[i+min];
cum_pi[min+i] = cumulative1;
A = cum_pi[min+i];
B = 1 - A;
w[min+i] = (i+min)* pi[min+i];    // mean value at different values of grey level
cumulative2 = cumulative2 + w[i+min];
cum_w[min+i] = cumulative2;
C = cum_w[min+i];
D = ((C*A) - C);
E = pow(D,2);  E=E*100;
F = A*B*100;
G = division(E,F);
 class_means[min+i] = G;

//   printf("%d\t%d\t%lf\t%lf\t%lf\t%lf\n", min+i , his_togram[min+i],  pi[min+i], A, w[min+i],G);
                                        }



/* The routine to search for the gerylevel that maximises the class_means */
/* and this value is the required treshold_value                          */



variance=0.0;
for (i=1; i<=(max-min); i++)  {

if (class_means[i+min]> variance) {
variance = class_means[min+i];

low_threshold = i+min;
high_threshold = max;

}

}

// printf("%d\t%d\n\n", low_threshold, high_threshold);

    }
```

```
// FILENAME: C:/VISION_P/CONTROL/MINFINDA.RSC


/**

      COMMENT: This program is used to read the first and
      last pixel of a finge during an interferogram line scan

      PROGRAM: MINAMA_FIND
      BY: ADE IDOWU (8/6/98)
      VERSION: 4.0
// #include "c:/vision_p/minima7a.h"



// Definition of data structure

struct minima
{   int val;
    int y;
    int x;
   };

 minima intensity[300];


mini_find(nfi, x_coord,y_scan_lenght,peak, position_start_y, position_end_y,
intensity_y,right_flag,left_flag)

{

// Definition of program variables
  int i, j, on_flag, temp;


// Read data fringe pixel data during line scan


for (i=1; i<y_scan_lenght; i++)  {
 intensity[i].y = i;
 intensity[0].x = x_coord;
 IpReadPix(nfi, intensity[0].x, intensity[i].y , &intensity[i].val);
//  printf("%d\t\t\t%d\t\t\t%d\t\t\t\n", intensity[0].x, intensity[i].y, intensity[i].val);
//  printf("%d\n",  intensity[i].y);
    printf("%d\n",  intensity[i].val);

                                                    }
// Set flag before fringe maxima detection

  on_flag = 1;
```

```
 peak = 0;


// Start fringe fringe maxima detection

 for(i=1; i<=y_scan_lenght; i++)

{

 if(intensity[i].val == 1)

    if(on_flag == 1)
      {
       peak = peak + 1 ;
       temp =   intensity[i].y;
       position_start_y[peak] = temp;
       temp =   intensity[i].val;
       intensity_y[peak] = temp;


                }
                                                    }
// Set  on_flag to 1 if a OFF pixel is detected

   on_flag = 0;



    if(intensity[i].val == 0)
     {
       on_flag = 1;
     temp =   intensity[i].y;
       position_end_y[peak] = (temp - 1);

    }


            }

right_flag = 0; left_flag = 0;

if (intensity[1].val == 1) left_flag = 1;
if (intensity[1].val == 1 && intensity[2].val == 1) left_flag = 1;

if (intensity[(y_scan_lenght - 1)].val == 1) right_flag = 1;
if (intensity[y_scan_lenght].val == 1) right_flag = 1;

//   printf("The detected minima peaks are: %d\n\n", peak);


}
```

```
/***** FILENAME: "C:\VISION_P\MATLAB_RSC\STORE_4A_.RSC" *****/

/****
        This program is used to store the output data
        of the measured interferogram data on disk.

        Every time a measurement is made the data is appended to the file.


        Written by:  Ade Idowu
        Date: 29/4/96
        Version 1.1
                                                        ***/


 storage4(filename,u,x,y,z)
{

// Definition of program variables
            auto fp;

// Open destination file and append data
            fp = fopen(filename, "a");

// open has failed !!  so send error maessge
            if (fp==0)  {
            RmShowError("Cannot open file");
            return 0;
             }

// else opening file has succeeded

fprintf(fp, "%d\n", u);
fprintf(fp, "%d\n", x);
fprintf(fp, "%d\n", y);
fprintf(fp, "%d\n", z);

// Close destination file
            fclose(fp);


}
```

```
/***** FILENAME: "C:\VISION_P\MATLAB_RSC\STORED_5.RSC" *****/

/****
        This program is used to store the output data
        of the measured fringe maxima data on disk.

        Every time a measurement is made the data is appended to the file.

         Written by:  Ade Idowu
         Date: 29/4/96
         Version 1.1
                                                        ***/

 storage5(filename,u,v)
{

// Definition of program varuables
        auto fp;

// Open destination file and append data
        fp = fopen(filename, "a");
        if (fp==0)  {

// open has failed !!  so send error maessge
        RmShowError("Cannot open file");
        return 0;
        }

// else opening file has succeeded


fprintf(fp, "%d\n", u);
fprintf(fp, "%d\n", v);


// Close destination file
    fclose(fp);

}
```

```
/***** FILENAME: "C:\VISION_P\MATLAB_RSC\STORED_9.RSC" *****/

/****
        This program is used to store the output data
        of the measured fringe maxima data-points on disk.

        Every time a measurement is made the data is appended to the file.

        Written by:  Ade Idowu
        Date: 29/4/96
        Version 1.1
                                                        ***/


 storage9(filename,u)
{

// Definition of program varibles
        auto fp;

// Open destination file and append data
        fp = fopen(filename, "a");


// open has failed !!  so send error maessge
        if (fp==0)  {
        RmShowError("Cannot open file");
        return 0;
  }

// else opening file has succeeded

fprintf(fp, "%d\n", u);

// Close destination file
        fclose(fp);


}
```

```
/************ FILENAME: C:\VISION_P\MATLAB_RSC\SCALED_1.RSC ******************/

  /**

        PROGRAM DESCRIPTION: This program is used to scale (re-map) the AOI
        of a detected interferogram, in oder to increase its spatial
        resolution -

        DEVELOPED BY: ADE IDOWU  (20/2/98)
        Version 1.0

                        **/


scaling(filename,rotate_angle)

{

// Definition of program variables
            int origin[2], num, cut_1[2], paste_orig[2], sub_orig[2];
            int square_zone[2];


// Define origin of region containing interferogram
// - old AOI
            origin[0] = 109;
            origin[1] = 94;

// Define new AOI size of interferogram
            cut_1[0] = 52;
            cut_1[1] = 80;

// Define origin of new origin of the old AOI
            paste_orig[0] = 128 - (cut_1[0]/2);
            paste_orig[1] = 128 - (cut_1[1]/2);

// Define new AOI size of interferogram
            square_zone[0] = 30;
            square_zone[1] = 30;

// Define origin of the new AOI
            sub_orig[0] = paste_orig[0] +((cut_1[0] - square_zone[0])/2) ;
            sub_orig[1] = paste_orig[1] +((cut_1[1] - square_zone[1])/2) ;

// Create 'dummy' space forouput images
            subtract("input", "input", "new");
            subtract("input", "input", "new1");

// Extract old AOI to 'c' and paste it
// to a 256 by 256 spatial resolution
```

```
        cut(filename,origin,cut_1, "c");
        paste("c", "new",paste_orig);

// Scale old AOI to a 256 by 256 spatial resolution
        IpSftzm("new", "new1",paste_orig ,3,V_4NEIGH);

// Create 'dummy' space for output image
        copy("new1", "new2");

// Create 'dummy' space forouput images
        subtract("input", "input", "newer");
        subtract("input", "input", "newer1");

// Extract old AOI to 'c' and paste it
// to a 256 by 256 spatial resolution
        cut("new",sub_orig,square_zone, "d");
        paste("d", "newer",sub_orig);

// Create 'dummy' space for output image
        IpSftzm("newer", "newer1",sub_orig ,9,V_4NEIGH);

// Create 'dummy' space for output image
        copy("newer1", "new3");

// Rotation("newer", "newer1", rotate_angle, {128,128}, 1);


}
```

```
#include "c:/vision_p/sin_fit2.rsc"

// FILENAME: "C:/VISION_P/MATLAB_RSC/PHASE2.RSC"

/**

        COMMENT: This program is used to compute maximium point of a
        sinusoid fit.

        BY: ADE IDOWU (22/2/98)
                                                **/

phased()
{

// Definition of program variables

        int i,x;
        float intensity_e[400],spacing,amplitude,phase,dc;
        int n;
        float intensity_t[400], pi, max_point,max_intensity;
        pi = acos(-1);

// Scan interferogram for intensity data
for (i=1; i<256; i++)
{
        x =128;
        IpReadPix("new_rotate", x, i, &intensity_e[i]);
        //printf("%d\t\t%lf\n", i, intensity_e[i]);
}

// Apply sinusodial fit
n = 256; spacing = 58;

        sinusoid_fit(intensity_e,n,spacing, &phase, &amplitude, &dc);
        //printf("Fringe phase: %lf\n", phase);
        //printf("Fringe amplitude: %lf\n", amplitude);
        //printf("dc opffset: %lf\n", dc);

// Define the fitted interferogram sinusodial function:
for (i=1; i<256; i++)
        {
        intensity_t[i] = dc + amplitude*cos( (((2*pi)/spacing)*i) + phase);
        }

// Routine to find the maximum point of the interferogram function
```

```
max_intensity = 0.0;
for (i=1; i<=256; i++) {

        if (intensity_t[i] > max_intensity) {
                max_intensity = intensity_t[i];
                max_point = i;
}
}

        printf("Fringe maximum point: %lf\n",max_point);
}
```

```
/************ FILENAME: C:\VISION_P\COORDS.RSC ******************/

  /**

        PROGRAM DESCRIPTION: This program is used to calculate
        the required coordinates of an interferogram line scan

        DEVELOPED BY: ADE IDOWU  (23/2/98)
        Version 1.0

                        **/
find_coords(nfi,x_coord,y_coord,rotation_angle,b_x,b_y,a_x,a_y)

{
float pi, angle_radians,cot;
int i, j, on_flag, temp;
int num, value[200];

        pi = acos(-1);     // PI constant defined
        //cot = cos(angle_radians)/sin(angle_radians);

if (rotation_angle < 90 && rotation_angle >= 45 )
        {
        angle_radians = ((90 - rotation_angle)/180) * pi;
        a_x = (x_coord/2) - (y_coord/2)*tan(angle_radians);
        a_x = ceil(a_x);
        a_y = y_coord;
        b_x = (x_coord/2) + (y_coord/2)*tan(angle_radians);
        b_x = ceil(b_x);
     if(rotation_angle == 45) a_x = 1;
        b_y = 1;
        }

if (rotation_angle >= 90 && rotation_angle <= 135)
        {
        angle_radians = ((rotation_angle - 90)/180) * pi;
        a_x = (x_coord/2) + (y_coord/2)*tan(angle_radians);
        a_x = ceil(a_x);
        a_y = y_coord;
        b_x = (x_coord/2) - (y_coord/2)*tan(angle_radians);
        b_x = ceil(b_x);
     if(rotation_angle == 135) b_x = 1;
        b_y = 1;
        }

if (rotation_angle > 135 && rotation_angle <= 180)
        {
        angle_radians = ((180 - rotation_angle)/180) * pi;
        a_y = (y_coord/2) + (x_coord/2)*tan(angle_radians);
        a_y = ceil(a_y);
        a_x = x_coord;
        b_y = (y_coord/2) - (x_coord/2)*tan(angle_radians);
        b_y = ceil(b_y);
        b_x = 1;
```

```
            }

if (rotation_angle < 45 && rotation_angle >= 0)
            {
            angle_radians = ((rotation_angle)/180) * pi;
            a_y = (y_coord/2) + (x_coord/2)*tan(angle_radians);
            a_y = ceil(a_y);
            a_x = 1;
            b_y = (y_coord/2) - (x_coord/2)*tan(angle_radians);
            b_y = ceil(b_y);
            b_x = x_coord;
            }

//printf("Value cot is: %f\n", cot);
//printf("Value a_x is: %f\n", a_x);
//printf("Value a_y is: %f\n", a_y);
//printf("Value b_x is: %f\n", b_x);
//printf("Value b_y is: %f\n", b_y);

}
```

PROGRAM LISTINGS 2: ALGORITHMS ARE USED TO CHARACTERISE THE PERFORMANCE OF THE INTERFEROMETER X AND Y SERVOS

```
// FILENAME: C:/VISION_P/BACK_1D.RSC

/****

        NOTE: This program samples the fringe pattern at servo pulse signals
        of 1 and quantifies the amount of backlash of the servo WITH the
        DITHER signal for the UP and DOWN direction @ specified frequency and an amplitude


    PROGRAM: 'FRINGE_ARQUIRE
    DEVELOPED BY : ADE IDOWU
    VERSION: 2.0 (23/11/96)

                                        ******/



#include "d:/vision_p/minbin2l.rsc"
#include "d:/vision_p/store.rsc"
#include "d:/vision_p/store_1.rsc"
#include "d:/vision_p/time_1.h"
#include "d:/vision_p/c5_servo.rsc"
#define LEFT   48
#define RIGHT 32
#define UP  128
#define DOWN  192
#define LOOP 10



 acquire_fringe()

   {

// Definition of program variables

    int temp, i, j, on_time;
    char result[200], a[200],b[200], c[200];
```

```
// Create read/write files

for (i=1; i<=5; i++)
{
for (j=1; j<=10; j++)
{

  a[j] = "d:/vision_p/images/bk1_d";
  temp = j;
  b[j] =  temp;
  c[j] = i;

  strncat(a[j], b[j], 1000000);
  strncat(a[j], c[j], 1000000);
  result[j] = a[j];


// Read interferogram frame

 snap("temp", 1);
 save("temp", result[j], 3);
 on_time = 1;

// Actuate servo for 1 second and delay for 2 seconds
 timming(DOWN,on_time);
 timed(2);
 }


// Create read/write files

for (j=11; j<=20; j++)
{
  a[j] = "d:/vision_p/images/bk1_d";
  temp = j;
  b[j] =  temp;
  c[j] = i;

  strncat(a[j], b[j], 1000000);
  strncat(a[j], c[j], 1000000);
  result[j] = a[j];


// Read interferogram frame

 snap("temp", 1);
 save("temp", result[j], 3);
 on_time = 1;

// Actuate servo for 1 second and delay for 2 seconds
 timming(DOWN,on_time);
 timed(2);
                              }
```

```
// Create read/write files
for (j=21; j<=30; j++)
{

  a[j] = "d:/vision_p/images/bk1_d";
  temp = j;
  b[j] =  temp;
  c[j] = i;

  strncat(a[j], b[j], 1000000);
  strncat(a[j], c[j], 1000000);
  result[j] = a[j];


// Read interferogram frame
 snap("temp", 1);
 save("temp", result[j], 3);
 on_time = 1;

// Actuate servo for 1 second and delay for 2 seconds
 timming(UP,on_time);
 timed(2);
                              }


// Create read/write files

for (j=31; j<=40; j++)
{
  a[j] = "d:/vision_p/images/bk1_d";
  temp = j;
  b[j] =  temp;
  c[j] = i;

  strncat(a[j], b[j], 1000000);
  strncat(a[j], c[j], 1000000);
  result[j] = a[j];


// Read interferogram frame
 snap("temp", 1);
 save("temp", result[j], 3);
 on_time = 1;

// Actuate servo for 1 second and delay for 2 seconds
 timming(UP,on_time);
 timed(2);
                              }


}
     }
```

```
#include "d:/vision_p/time.h"
#include "d:/vision_p/resolve5.h"
#define LOOP 60
#define ANGLE_COUNT 4000

// FILENAME: C:/VISION_P/FRIN_AC3.RSC

/****

      PROGRAM: 'FRINGE_ARQUIRE 3'
      DEVELOPED BY : ADE IDOWU
      VERSION: 1.0 (6/12/96)
NOTE: This program samples the fringe pattern at specifed
angular intervals of n degress.

                                                ******/


#include "d:/vision_p/time.h"
#include "d:/vision_p/resolve5.h"
#define LOOP 60
#define ANGLE_COUNT 4000

new_sub()
{
resolver_ac();
}



resolver_ac()

{


    int j,i,angular_value,temp;
    char result[200], a[200], b[200];

for (j=1; j<=LOOP; j++)
{

  a[j] = "d:/vision_p/images/n1_";
  temp = j*(360/LOOP);
  b[j] =  temp;
```

```
 strncat(a[j], b[j], 1000000);
 result[j] = a[j];

}


for (j=1; j<=LOOP; j++)
{

 for (i=1; i<=ANGLE_COUNT; i++)
{
   temp = j*(360/LOOP);
 angular_value = abs(resolver());


  if(angular_value ==temp) {
  snap("temp", 1);
  break;   }

        }

save("temp", result[j],3 );


}

  temp = LOOP;
  return temp;

}
```

```
/************ FILENAME: C:\VISION_P\CONTROL.RSC ******************/

  /**

        PROGRAM DESCRIPTION: This program is used to obtain SYSTEM IDENTIFICATION
        data for interferometer X-Y servo

        DEVELOPED BY: ADE IDOWU  (2/9/98)
        Version 5.0

                        **/

#include "c:/vision_p/maxima_1.rsc"
#include "c:/vision_p/tre_new2.rsc"
#include "c:/vision_p/minfinda.rsc"
#include "c:/vision_p/stored_1.rsc"
#include "c:/vision_p/stored_2.rsc"
#define MAGIC 0x12345678     // Use a specified magic number (given in VISILOG program manual)



// Define program data structure
struct valued
{       int fm;
        float fd;
        int fo;
        float fs;
};

valued fringedata[200];

test()
 {


// Definition of program variables
        int g, h[20], k[20], j, i;
        float l, count[200],m,temp;
        int loop_1,loop_2;
        char file[100],result[200], a[200],b[200], c[200],d[200],e[200],f[200], bin_file[200];
        loop_1 = 5;
        loop_2 = 40;

// Definition of read/write files

 for (i=1; i <=loop_1 ; i++)  {
        a[i] = "c:/vision_p/model_X/modelx_";
        b[i] = i;
        c[i] =".txt";
```

```
strncat(a[i], b[i], 10000000);
strncat(a[i], c[i], 10000000);
file[i] = a[i];

        a[i] = "c:/vision_p/model_X/modelx_";
        b[i] = i;
        c[i] = ".bin";

strncat(a[i], b[i], 10000000);
strncat(a[i], c[i], 10000000);
bin_file[i] = a[i];

storage1(file[i]);

 for (j=1; j <=loop_2 ; j++)  {
    d[j] = "c:/vision_p/model_X/bak2_";
    e[j] =  j;
    f[j] = i;
strncat(d[j], e[j], 10000000);
strncat(d[j], f[j], 10000000);
result[j] = d[j];

count[j] = j;

   mini_bin(result[j], &g, &h, &k,&l, &m);
   storage(file[i],count[j], g,l,m);

 //Output data are:
temp =    count[j];
fringedata[j].fm = temp;    // Read fringe measurment no. - fm
fringedata[j].fd = g;                    // Read fringe density - fd
fringedata[j].fo = l;                    // Read fringe orientation - fo
fringedata[j].fs = m;                    // Read fringe spatial freq.- sf

        }

// Save data as a binary file

RmSaveVal(bin_file[i],fringedata, MAGIC);    //Save data of arrays in structure


  }
    }
```

```
// Function used acquire fringe data from an interferogram

 mini_bin(file1, a, b, c, d, e)
```

```
{

// Definition of program variables

double angle_degrees,rotation_angle;
int high_threshold, low_threshold, level[2];
char filename;
int value, y[100], j, in[100];
int x, y_lenght, flag1,temp;
float spatial_frequency;

// Compute interferogram spatial frequency and orientation
angle_degrees = locating_pixels(file1, &flag1, &spatial_frequency);
e = spatial_frequency;
// if there is an infinite fringe spacing
if (flag1 <= 1)
{

printf("   ");
a=1;
b =0;
c= 0;
d = 45;

}

// if there is an finite fringe spacing but the fringe orientaion is 90 degress
if (flag1==2) {
if (angle_degrees==90)
    {
    d = 90;
    rotation_angle = 180;
    }

    // if there is an finite fringe spacing but the fringe orientaion is 0 degress
    if (angle_degrees== 0)  rotation_angle=d = 90.01;
    d = angle_degrees;
    // If the angle is positive, and not 0 or 90
    if (d > 0 && d != 0 && d !=90)  // If the angle is positive, and not 0 or 90
    rotation_angle = 90 - (angle_degrees);
    if (d < 0)
    {
d = 180 + d; // If the angle is negative
rotation_angle = 90 - (angle_degrees);
}
    // define filename
    filename = "drain_cut";

    // Evaluate the optimal threshold values of the interferogram
optimal_threshold(filename, &low_threshold, &high_threshold);
    level[0]  = low_threshold;
    level[1] =  high_threshold;
    Threshold(filename, level, "crab2");
```

```
IpSetAoiMode(1);
copy("crab2", "crab2_old");
addmask("crab2_old", "c:/vision_p/images/mask1");
copy("crab2_old", "crab2a");
IpSetAoiMode(0);

    // Rotate interferogram and select an area of interest
    rotation("crab2a", "crab3a", rotation_angle, {28,45}, 0);

    // Apply morphological function to imnterferogram
    Ldilate("crab3a",0,2, "crab4"  );

    // Read fringe pixel intensity at x and y locations
x=28; y_lenght=90;
mini_find("crab4", x, y_lenght, &value, &y, &in);

     // Read fringe maxima data
a = value;
if (value >= 1) {
     for(j=1; j>=value; j++)
     {
  temp  = y[j]; b[j] =temp;
  temp= in[j]; c[j] = temp;
                                          }

                                           }
                                            }
}
```

```
/***** FILENAME: "C:\VISION_P\CONTROL\STORED_1.RSC" *****/


/****
        This program is used to store the output data
        of the measured fringe  data on disk.

        Every time a measurement is made the data is appended to the file.


         Written by:  Ade Idowu
         Date: 29/4/96
         Version 1.1
                                                        ***/


 storage(filename,w,u,x,y,z)
{

// Definition of program varibles
   auto fp;

// Open destination file and append data
   fp = fopen(filename, "a");


// open has failed !!  so send error maessge
   if (fp==0)  {
   RmShowError("Cannot open file");
   return 0;
  }

// else opening file has succeeded

fprintf(fp, "%d\t\t%d\t\t%d\t\t%d\t\t%f\n" ,w,u ,x,y, z);

// Close destination file
     fclose(fp);


}
```

```
/***** FILENAME: "C:\VISION_P\MATLAB_RSC\STORED_2.RSC" *****/


/****
        This program is used to store the output data
        of the measured fringe maxima data-points on disk.

        Every time a measurement is made the data is appended to the file.

        Written by:  Ade Idowu
        Date: 29/4/96
        Version 1.1
                                                        ***/


 storage1(filename)
{

// Definition of program varibles
   auto fp;

// Open destination file and append data
   fp = fopen(filename, "a");

// open has failed !!  so send error maessge
   if (fp==0)  {
   RmShowError("Cannot open file");
    return 0;
  }

// else opening file has succeeded

fprintf(fp, "\n\n\n\n" );
fprintf(fp, "MEASURE_NUMBER  NO_OF_MININIMA  ORIENTATION   SPATIAL FREQ.\n");
fprintf(fp, "--------------  -----------   -------------   ------------ \n");


// Close destination file
     fclose(fp);


 }
```

```
// FILENAME: C:/VISION_P/CONTROL/MINFINDA.RSC


/**

     COMMENT: This program is used to read the first and
     last pixel of a finge during an interferogram line scan

     PROGRAM: MINAMA_FIND
     BY: ADE IDOWU (8/6/98)
     VERSION: 4.0
                                                    **/


// Definition of data structure

struct minima
{   int val;
    int y;
    int x;
   };

minima intensity[300];


mini_find(nfi, x_coord,y_scan_lenght,peak, position_start_y, position_end_y, intensity_y)

{

// Definition of program variables

  int i, j, on_flag, temp;


// Read data fringe pixel data during line scan


for (i=1; i<y_scan_lenght; i++)  {
 intensity[i].y = i;
 intensity[0].x = x_coord;
 IpReadPix(nfi, intensity[0].x, intensity[i].y , &intensity[i].val);

                                                    }
// Set flag before fringe maxima detection

  on_flag = 1;
  peak = 0;
```

```
// Start fringe fringe maxima detection

 for(i=1; i<=y_scan_lenght; i++)

{

 if(intensity[i].val == 1)

// Set  on_flag to 1 if a ON pixel is detected

    if(on_flag == 1)
      {
        peak = peak + 1 ;
        temp =   intensity[i].y;
        position_start_y[peak] = temp;
        temp =   intensity[i].val;
        intensity_y[peak] = temp;


// Set  on_flag to 1 if a OFF pixel is detected                    }

   on_flag = 0;



    if(intensity[i].val == 0)
     {
          on_flag = 1;
      temp =   intensity[i].y;
          position_end_y[peak] = (temp - 1);
     }

               }

//   printf("The detected minima peaks are: %d\n\n", peak);

    }
```

PROGRAM LISTINGS 3: SERVO CONTROL ALGORITHMS

```
/*************************************************************

 FILENAME : C:\VISION_P\CONTROL\CON_DATA.RSC

          *************************************************/

/*************************************************************************

 PROGRAM DESCRIPTION: This program is used to search for the
 accurate value of the maximum/minimum fringe density by using
 the detected array index value of the maximum spacial frequency.
 This program also evaluates the values of the spindle angular position
 and the interferogram  orientation at the detected index value.


 DEVELOPED BY: ADE IDOWU  (20/8/98)

 VERSION: 4.0

*************************************************************************/

#define MAGIC 0x12345678     // Use a specified magic number (given in VISILOG program manual)


//testing()

//{
//char filename;
//float inclination, modulus, average;

//filename = "c:/vision_p/control/con_data/cont_2.bin";

//readfringe(filename, &modulus, &inclination, &average);
//printf("The tilt correction signal modulus is\t %d\n\n\n\n", modulus);
//printf("The tilt inclination is\t %d\n\n\n\n", inclination);
//}
```

```
readfringe(filename, modulus, inclination, average)
{

// Definition of variables

int mag, array,i;
int temp,max_indicator;
int loop;
float maxsf, maxfd, maxfo,maxsp;
float minsf, minfd, minfo,minsp;
float dc_fd, r;  // DC fringe_density

// CARRYOUT INITIALISATIONS

    mag = MAGIC;
    loop = 60;

// Read fringe data from binary file
RmReadVal(filename, &array, &mag);     // Read data of arrays in structure


// Dispaly contents of the file (DISABLED)

//  for (i=1; i<=loop; i++)
//{
// printf("fm[%d] = %d\n",i, array[i].fm);  // Display first array in structure
// printf("fd[%d] = %d\n",i, array[i].fd);  // Display second array in structure
// printf("fo[%d] = %d\n",i, array[i].fo);  // Display third array in structure
// printf("fs[%d] = %d\n",i, array[i].fs);  // Display fourth array in structure
// printf("fa[%d] = %d\n",i, array[i].fa);  // Display fourth array in structure
//}


// FIND THE INDEX OF THE MAXIMUM SPATIAL FREQUENCY

 for (i=1; i<=loop; i++)
{
 if (array[i].fd >= temp) {
 temp = array[i].fd;
 max_indicator = i;
                    }

}
    maxsf = array[max_indicator].fs;   // MAXSF = MAXimum Spacial Frequency
    maxfd = array[max_indicator].fd;  // MAXFD = MAXimum Fringe Density
    maxfo = array[max_indicator].fo; // MAXFO = MAXimum Fringe Orientation
    maxsp = array[max_indicator].fa; // MAXSP =MAXimum Spindle Position

// Display output data (DISABLED)

    //printf("The index of maximum spacial frequency:\t%d\n", max_indicator);
    //printf("The value maximum spacial frequency:\t%4.2f\n", maxsf);
    //printf("The  maximum fringe density:\t%d\n", maxfd);
    //printf( "The fringe orientation for the MAXFD:\t%3.2f\n" , maxfo);
```

```
    //printf("The spindle position at the MAXFD:\t %d\n\n\n\n", maxsp);
    //printf(".....where MAXFD = MAXimum Fringe Density \n\n");



// FIND THE INDEX OF MINIMUM SPACIAL FREQUENCY

  temp = maxsf;
  for (i=1; i<loop; i++)
{
  if (array[i].fd < temp && array[i].fd > 0) { //can use .fs for more accuracy
  temp = array[i].fd;
  max_indicator = i;
                }

}


    minsf = array[max_indicator].fs;   // MAXSF = MAXimum Spacial Frequency
    minfd = array[max_indicator].fd;  // MINFD = MINimum Fringe Density
    minfo = array[max_indicator].fo; // MINFO = MINimum Fringe Orientation
    minsp = array[max_indicator].fa; // MINSP =MINimum Spindle Position

// Display output data (DISABLED)

    //printf("The index of minimum spacial frequency:\t%d\n", max_indicator);
    //printf("The value minimum spacial frequency:\t%4.2f\n", minsf);
    //printf("The  minimum fringe density:\t%d\n", minfd);
    //printf( "The fringe orientation for the MINFD:\t%3.2f\n" , minfo);
    //printf("The spindle position at the MINFD:\t %d\n\n\n\n", minsp);
    //printf(".....where MINFD = MINimum Fringe Density \n\n");


// Compute average fringe density
    dc_fd = (maxfd + minfd)/2;
    average = dc_fd;

// Compute modulus of the tilt required to be removed
    r = (maxfd - dc_fd);

// Computed values of modulus and oreintation required by the control sub-routine:
    modulus = r;
    inclination = maxsp;



}
```

/***** FILENAME: "C:\VISION_P\CONTROL\CONTROL_1.RSC" *****/

```
/****
        This program is for servo linear control


        Written by:  Ade Idowu
        Date: 29/10/96
        Version 4.0                                                    ***/


#include "c:/vision_p/maxima_1.rsc"
#include "c:/vision_p/servo/linear1.rsc"
#include "c:/vision_p/servo/recursive.rsc"
#include "c:/vision_p/servo/con_data.rsc"
#include "c:/vision_p/servo/random.rsc"
#define RIGHT 32
#define LEFT 48
#define UP 128
#define DOWN 192



servo_control()

{

float m_x, c_x, m_y, c_y, modulus, inclination, average_fd, new_fd;
float r_x, r_y, error, int_error, tolerance, counts;
float pmx[200], pmy[200], interval, temp, errorx[200], errory[200];
float emx[200], emy[200], datax[200], datay[200], sumx, sumy;
float integy[200], integx, rx,ry, dc_value, input_x, input_y;
char filename, system_id;

tolerance = 1e-3;
counts =10;
system_id = "S"

// Calculate values of control servo signal modulus and orientation
```

```
// and average fringe density
control_data(filename, &modulus, &inclination, &average_fd);


// Evaluate servo dynamic coefficients
if (system_id == "R") {
// Calulate values of function slope and y-intercept using Recursive least square akgorithm
// where the dc value is a value known a piori
dc_value = 2;

system_id2(filename, dc_value,  &m_x, &c_x);
system_id2(filename, dc_value, &m_y, &c_y);

// Calculate othogonal control signals
r_x = xcontrol_signal(m_x,c_x,error, inclination);
r_y = ycontrol_signal(m_y,c_y,error, inclination);

// Apply control signals to interferometer
control_interferometer(r_x, r_y, counts);

// Read new value of fringe density
random_read(filename, &new_fd);

// Calculate new control signal modulus
modulus = modulus - new_fd;
}

if (system_id == "S") {

// Calulate values of function slope and y-intercept using Sinple least square akgorithm
dc_value = 0;
system_id(filename, &m_x, &c_x);
system_id(filename, &m_y, &c_y);
}


while (error < tolerance)

{

// Calculate othogonal control signals
r_x = xcontrol_signal(m_x,c_x,modulus, inclination);
r_y = ycontrol_signal(m_y,c_y,modulus, inclination);

// Apply control signals to interferometer
control_interferometer(r_x, r_y,counts);

// Read new value of fringe density
random_read(filename, &new_fd);


// Calaculate errro signal
error = (average - new_fd);
```

```
// Calculate new control signal modulus
modulus = error;

}



// Function to calculate x-axis control signal

xcontrol_signal(m, c, modulus, inclination)
{

float value, temp;

 temp = ((modulus*cos(inclination)) - c);
 value = division(temp, m);
 return value;

}



// Function to calculate y-axis control signal

ycontrol_signal(m, c, modulus, inclination)
{

float value, temp;

 temp = ((modulus*sin(inclination)) - c);
 value = division(temp, m);
 return value;

}



// Function to control interferometer

control_interferometer(rx, ry,counts))
{
float delta_rx, delta_ry;

delta_rx = division(abs(rx),counts);
delta_ry = division(abs(ry),counts);

// Case 1: both rx and ry are POSITIVE
```

```
if (rx > 0 && ry > 0)

{

for (i=1; i<= counts; i++)
{
timmming(RIGHT,delta_rx)
timmming(UP,delta_ry)
}


}

// Case 2: both rx and ry are NEGATIVE

if (rx < 0 && ry < 0)

{

for (i=1; i<= counts; i++)
{
timmming(LEFT,delta_rx)
timmming(DOWN, delta_ry)
}


}

// Case 3: both rx is POSITIVE and ry is NEGATIVE

if (rx > 0 && ry < 0)

{

for (i=1; i<= counts; i++)
{
timmming(RIGHT, delta_rx)
timmming(DOWN,delta_ry)
}


}


// Case 3: both rx is NEGATIVE and ry is POSITIVE

if (rx > 0 && ry < 0)

{

for (i=1; i<= counts; i++)
{
timmming(LEFT, delta_rx)
```

```
timmming(UP, delta_ry)
}


}


}
```

```
/***** FILENAME: "C:\VISION_P\CONTROL\LINEAR_1.RSC" *****/

/****
        This program is for servo linear fit to servo SYSTEM ID data


        Written by:  Ade Idowu
        Date: 29/10/97
        Version 4.0
                                                                ***/

#define MAGIC 0x12345678        // Use a specified magic number (given in VISILOG program manual)
#define N 10                    // Total number of fringe data points
#include "c:/vision_p/control/servo_1.rsc"
#include "c:/vision_p/finalg.h"


// Testing program (DIABLED)

//testing()

//{

//char filename;
//float slope, y_intercept;

//filename = "c:/vision_p/control/con_data/cont_2.bin";

//system_id(filename, &slope, &y_intercept);

//}



system_id(filename, slope, y_intercept)

{


// Define program variables
float x_data[200], y_data[200], c,m;
float temp;
int loop;
int mag, array,i;

mag = MAGIC;
loop = N;

// Read fringe data from binary file
```

```
RmReadVal(filename, &array, &mag);     // Read data of arrays in structure

 for (i=1; i<=loop; i++)
{
printf("fd[%d] = %d\n",i, array[i].fd);  // Display second array in structure
temp = array[i].fd;
x_data[i] = i;
y_data[i] = temp;
}

// Compute servo dynamic coefficients for linear fit
linear_fit(x_data, y_data, &m, &c);
printf("Value of slope: %f\n\n", m);
printf("Value of Y-intercept: %f\n\n", c);
slope = m;
y_intercept = c;

}




// Least sqaure fit routine


linear_fit(x_data,y_data, m,c)
{

// Definition of program variables

int n, i;
float x[200], y[200], squared[200],mean_x, mean_y, sum_x, sum_y;
float sum_sqx, product_xy, xy[200], temp1, temp2;



sum_x = 0;
sum_y = 0;
sum_sqx = 0;
product_xy = 0;
n = N;
for (i=1; i<=n; i++)
{

temp1 = x_data[i];
x[i] = temp1;
temp1 = y_data[i];
y[i] = temp1;
printf("x[%d] = %f\n",i, x[i]);
```

```
printf("y[%d] = %f\n",i, y[i]);

}

temp1 = 0;


for (i=1; i<= n; i++)
{


sum_x = sum_x + x[i];
sum_y = sum_y + y[i];
temp1 = x[i]*x[i];
squared[i] = temp1;
xy[i] = x[i]*y[i];
sum_sqx = sum_sqx + squared[i];
product_xy = product_xy + xy[i];


}


mean_x = division(sum_x, n);
mean_y = division(sum_y, n);



// Compute Slope

temp1 = ((n*product_xy) - (sum_x*sum_y));
temp2 = (n*sum_sqx) - (sum_x*sum_x);

m = division(temp1,temp2);


// Compute Y-intercept

c = mean_y - (m*mean_x);




}
```

```
/************ FILENAME: C:\VISION_P\SERVO\RANDOM.RSC ******************/

  /**

        PROGRAM DESCRIPTION: This program is used to randomly searc/read
        the fringe density (on-the-fly) at specified angular postion


        DEVELOPED BY: ADE IDOWU  (23/11/98)
        Version 1.1

                        **/

#include "c:/vision_p/maxima_1.rsc"
#include "c:/vision_p/tre_new2.rsc"
#include "c:/vision_p/minfindf.rsc"
#include "c:/vision_p/control/stored_1.rsc"
#include "c:/vision_p/stored_2.rsc"
#include "d:/vision_p/time.h"
#include "d:/vision_p/resolve5.h"
#define MAGIC 0x12345678     // Use a specified magic number (given in VISILOG program manual)



random_read(inclination, &measured_fd)
 {


// Definition of program variables
            int g, h[20], k[20];
            float l, m;

// Search randomly, and acquire fringe density value at
// specified angular position

   if (resolver() == inclination)
  {
mini_bin(result[j], &g, &h, &k,&l, &m);
measured_fd = g;
break;
  }
     }
```

```
// Function used acquire fringe data from an interferogram

 mini_bin(file1, a, b, c, d, e)

{


// Definition of program variables
double angle_degrees,rotation_angle;
int high_threshold, low_threshold, level[2];
char filename;
int value, y[100], j, in[100];
int x, y_lenght, flag1,temp;
float spatial_frequency;

// Compute interferogram spatial frequency and orientation

        angle_degrees = locating_pixels(file1, &flag1, &spatial_frequency);
        e = spatial_frequency;
        // if there is an infinite fringe spacing
        if (flag1 <= 1)
        {
        printf("   ");
        a=1;
        b =0;
        c= 0;
        d = 45;
        }
        // if there is an finite fringe spacing but the fringe orientaion is 90 degress
        if (flag1==2) {
        if (angle_degrees==90)
        {
        d = 90;
        rotation_angle = 180;
        }

        // if there is an finite fringe spacing but the fringe orientaion is 0 degress
        if (angle_degrees== 0)  rotation_angle=d = 90.01;
        d = angle_degrees;
        // If the angle is positive, and not 0 or 90
        if (d > 0 && d != 0 && d !=90)  // If the angle is positive, and not 0 or 90
        rotation_angle = 90 - (angle_degrees);
        if (d < 0)
        {
        d = 180 + d; // If the angle is negative
        rotation_angle = 90 - (angle_degrees);
        }


        // define filename
        filename = "drain_cut";
optimal_threshold(filename, &low_threshold, &high_threshold);
```

```
level[0] = low_threshold;
    level[1] = high_threshold;
    Threshold(filename, level, "crab2");
IpSetAoiMode(1);
copy("crab2", "crab2_old");
addmask("crab2_old", "c:/vision_p/images/mask1");
copy("crab2_old", "crab2a");
IpSetAoiMode(0);


    // Rotate interferogram and select an area of interest
    rotation("crab2a", "crab3a", rotation_angle, {28,45}, 0);
    // Apply morphological function to imnterferogram
Ldilate("crab3a",0,2, "crab4"  );

    // Read fringe pixel intensity at x and y locations
x=28; y_lenght=90;
mini_find("crab4", x, y_lenght, &value, &y, &in);
// Read fringe maxima data
    a = value;
if (value >= 1) {
    for(j=1; j>=value; j++)
    {
temp  = y[j]; b[j] =temp;
temp= in[j]; c[j] = temp;
                                        }

                                         }
                                          }

}
```

```
/***** FILENAME: "C:\VISION_P\MATLAB_RSC\TIME.RSC" *****/

/****
        This program is used to apply timed inputs to the interferometer servo


        Written by:  Ade Idowu
        Date: 29/4/96
        Version 1.1                                                    ***/


//example()
//{

//   timming(10);  /* Input time in seconds */

//}


timming(direction, time)
 {
    int goal, wait;
    long high, low;

/* Apply the pulse for time seconds */

goal = (time*1000)+clock();
while(goal > clock()) {
 wait =clock();


IpMytask2(direction); }


}

}
```

```
/***** FILENAME: "C:\VISION_P\MATLAB_RSC\RESOLVE5.RSC" *****/

/****
        This program is used to the spindle angular position measured by the resolver


            Written by:  Ade Idowu
            Date: 12/6/96
            Version 3.0
                                                                    ***/

#include "d:/vision_p/data_re.h"


resolver()
 {

    double angle, temp1, temp2,temp3, re_low;
    int i, high, low;



IpResolve(&high,&low);
 re_low = lut_resolve(low);
temp1 = high *255;
temp2 = ((temp1) + re_low);
IpDivide(temp2,  65535, &temp3);
angle = temp3*360;

    return angle;
  }
```

```
// FILENAME: C:/VISION_P/CONTROL/MINFINDF.RSC


/**

    COMMENT: This program is used to read the first and
    last pixel of a finge during an interferogram line scan

    PROGRAM: MINAMA_FIND
    BY: ADE IDOWU (8/6/98)

    VERSION: 4.0
                                                    **/

// Definition of data structure

struct minima
{   int val;
   int y;
   int x;
  };

minima intensity[300];



mini_find(nfi, x_coord,y_scan_lenght,peak, position_start_y,intensity_y)

{

// Definition of program variables

 int i, j, on_flag, temp;


// Read data fringe pixel data during line scan

for (i=1; i<y_scan_lenght; i++)  {
 intensity[i].y = i;
 intensity[0].x = x_coord;
 IpReadPix(nfi, intensity[0].x, intensity[i].y , &intensity[i].val);
//  printf("%d\t\t\t%d\t\t\t%d\t\t\t\n", intensity[0].x, intensity[i].y, intensity[i].val);
//    printf("%d\n",  intensity[i].val);

                                                }
// Set flag before fringe maxima detection

 on_flag = 1;
 peak = 0;

// Start fringe fringe maxima detection

 for(i=1; i<=y_scan_lenght; i++)
```

```
{

 if(intensity[i].val == 1)

    if(on_flag == 1)
      {
        peak = peak + 1 ;
        temp =   intensity[i].y;
        position_start_y[peak] = temp;
        temp =   intensity[i].val;
        intensity_y[peak] = temp;

                    }

on_flag = 0;



 if(intensity[i].val == 0)
  {
        on_flag = 1;
    temp =   intensity[i].y;


   }

              }

//   printf("The detected minima peaks are: %d\n\n", peak);


   }
```

```
/***** FILENAME: "C:\VISION_P\CONTROL\CONTROL_2.RSC" *****/

/****
        This program is for servo ADAPTIVE control


        Written by:  Ade Idowu
        Date: 29/10/97
        Version 4.0
                                                                ***/

#include "c:/vision_p/finalg.h"
#include "c:/vision_p/servo/recursive.rsc"
#include "c:/vision_p/servo/linear1.rsc"
#include "c:/vision_p/servo/con_data.rsc"
#include "c:/vision_p/servo/random.rsc"
#define RIGHT 32
#define LEFT 48
#define UP 128
#define DOWN 192



servo_control()
{

tolerance = 1e-3;
counts =10;
system_id = "S"

float m_x, c_x, m_y, c_y, modulus, inclination, average_fd, new_fd;
float r_x, r_y, error, int_error, tolerance, counts;
float pmx[200], pmy[200], interval, temp, errorx[200], errory[200];
float emx[200], emy[200], datax[200], datay[200], sumx, sumy;
float integy[200], integx, rx,ry, dc_value, input_x, input_y;
float pmx[200], pmy[200];
char filename, system_id;


// Calculate values of control servo signal modulus and orientation
// and average fringe density
control_data(filename, &modulus, &inclination, average_fd);

// Evaluate servo dynamic coefficients
if (system_id == "R") {
// Calulate values of function slope and y-intercept using Recursive least square akgorithm
// where the dc value is a value known a piori
dc_value = 2;
```

```
system_id2(filename, dc_value,  &m_x, &c_x);
system_id2(filename, dc_value, &m_y, &c_y);

// Calculate othogonal control signals
r_x = xcontrol_signal(m_x,c_x,error, inclination);
r_y = ycontrol_signal(m_y,c_y,error, inclination);

// Apply control signals to interferometer
control_interferometer(r_x, r_y);

// Read new value of fringe density
random_read(filename, &new_fd);

// Calculate new control signal modulus
modulus = modulus - new_fd;
}

if (system_id == "S") {

// Calulate values of function slope and y-intercept using Sinple least square akgorithm
dc_value = 0;
system_id(filename, &m_x, &c_x);
system_id(filename, &m_y, &c_y);

}

// Get the REFERENCE and PHYSICAL outputs to converge

// Define adaptive control parameters
lamda = -0.50;
interval = 0.5;
counts = division(modulus, interval);

sum =0;
for(i =1; i<=abs(counts); i++)

{

// Calculate othogonal control signals
r_x = xcontrol_signal(m_x,c_x,modulus, inclination);
r_y = ycontrol_signal(m_y,c_y,modulus, inclination);


// Define adaptive control inputs
input_x = (interval*cos(inclination)*i);
input_y = (interval*sin(inclination)*i)

// Firstly define the REFERENCE outputs
pmx[i] = (input_x*m_x) +c_x;
pmy[i] = (input_y*m_y) +c_y;

// Secondly define the PHYSICAL outputs
pmx[i] = new_fd*cos(inclination);
pmy[i] = new_fd*sin(inclination);
```

```
// Thirdly define the error b/w PHYSICAL and REFERENCE for X-Y servos
temp = pmx[i] - pax[i];
errorx[i] = temp;
temp = pmy[i] - pay[i];
errory[i] = temp;

// Solve the adaptation law - using integration by Trapezium method
temp = errorx[i]*pmx[i];
datax[i] = temp
temp = errory[i]*pmy[i];
datay[i] = temp
sumx = sumx + interval*0.5*(datax[i] + datax[i+1]);
sumy = sumy + interval*0.5*(datay[i] + datay[i+1]);
integx[i] = sum;
integy[i] = sum;
rx = input_x*integx[i]*lamda;
ry = input_y*integx[i]*lamda;

// Control PHYSICAL X-Y servo
control_interferometer(rx, ry);
random_read(inclination, &new_fd); // Read new value of fringe density


// At the end of this loop it is anticipated that the PHYSICAL and REFERENCE outputs converge

}


// Check for convergence and apply 'fine' control
error = abs(average_fd - new_fd);
while (error < tolerance)

{

// Calulate values of function slope and y-intercept
system_id2(filename, error,  &m_x, &c_x);
system_id2(filename, error, &m_y, &c_y);


// Calculate othogonal control signals
r_x = xcontrol_signal(m_x,c_x,error, inclination);
r_y = ycontrol_signal(m_y,c_y,error, inclination);

// Apply control signals to interferometer
control_interferometer(r_x, r_y);


// Read new value of fringe density
random_read(filename, &new_fd);

// Calculate new control signal modulus
```

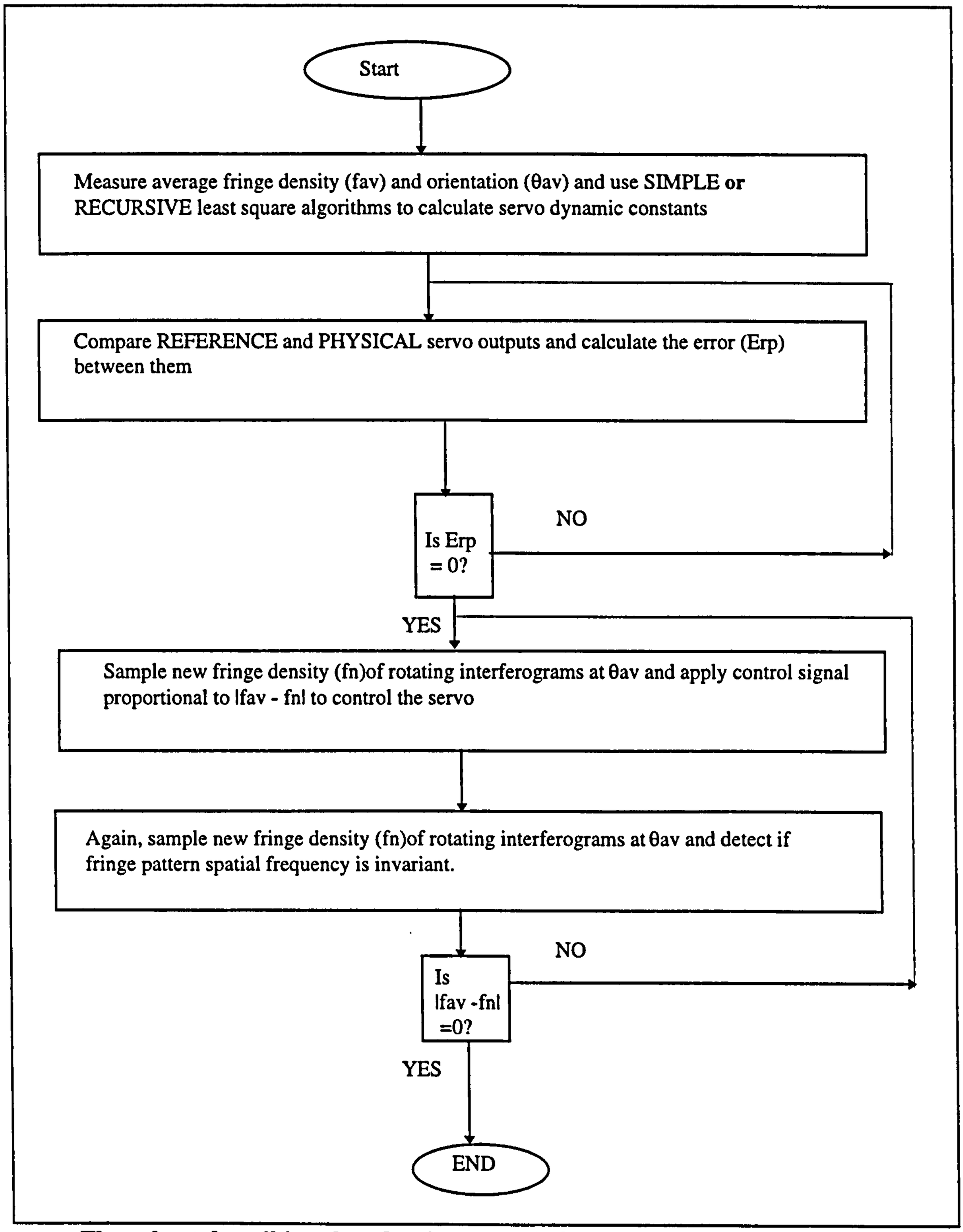
```
error = abs(average_fd - new_fd);

}
```

```
// Function to calculate x-axis control signal

xcontrol_signal(m, c, modulus, inclination)

{

float value, temp;

 temp = ((modulus*cos(inclination)) - c);
 value = division(temp, m);
 return value;

 }
```

```
// Function to calculate y-axis control signal

ycontrol_signal(m, c, modulus, inclination)

{

float value, temp;

 temp = ((modulus*sin(inclination)) - c);
 value = division(temp, m);
 return value;

 }
```

```
// Function to control interferometer

control_interferometer(rx, ry)

{

// Case 1: both rx and ry are POSITIVE
```

```
if (rx > 0 && ry > 0)

{



timmming(RIGHT,abs(rx))
timmming(UP,abs(ry))
}


// Case 2: both rx and ry are NEGATIVE

if (rx < 0 && ry < 0)

{
timmming(LEFT,abs(rx))
timmming(DOWN, abs(ry))
}



// Case 3: both rx is POSITIVE and ry is NEGATIVE

if (rx > 0 && ry < 0)

{
timmming(RIGHT,abs(rx))
timmming(DOWN,abs(ry))
}



// Case 4: both rx is NEGATIVE and ry is POSITIVE

if (rx > 0 && ry < 0)

{
timmming(LEFT,abs(rx))
timmming(UP,abs(ry))
}


}
```



Flow chart describing the adaptive control algorithm

```
/***** FILENAME: "C:\VISION_P\CONTROL\RECURSIVE.RSC" *****/

/****
        This program is for servo RECURSIVE linear fit to servo SYSTEM ID data


        Written by:  Ade Idowu
        Date: 29/10/97
        Version 4.0
                                                                ***/

#define MAGIC 0x12345678        // Use a specified magic number (given in VISILOG program manual)
#define N 10                    // Total number of fringe data points
#include "c:/vision_p/control/servo_1.rsc"
#include "c:/vision_p/servo/random.rsc"
#include "c:/vision_p/finalg.h"


// Testing program

system_id2(filename, new_fd, m,c);

{


// Definition of program variables

char filename;
float x_data[200], y_data[200], c,m, scalar;
float x_new[200], y_new[200], value, temp1;
float temp,n, new_fd;
int loop;
int mag, array,i, j;

 mag = MAGIC;
 loop = N;

// Read fringe data from binary file
//filename = "c:/vision_p/control/con_data/cont_2.bin";
RmReadVal(filename, &array, &mag);     // Read data of arrays in structure
```

```
 for (i=1; i<=loop; i++)
{
printf("fd[%d] = %d\n",i, array[i].fd);  // Display second array in structure
temp = array[i].fd;
x_data[i] = i;
y_data[i] = temp;
}

// Append new data-point to current data set

loop = N+1;
random_read(filename, &new_fd);
y_data[loop] = new_fd;
value = y_data[loop];
x_data[loop] = loop;
flag = 0;


// Sort data-ponts in ascending order

for(j = 2; j<=loop; j++)
{
        scalar = y_data[j];
        i = j - 1;
        while (y_data[i] > scalar && i>0)
        {
                temp1 = y_data[i];
                y_data[i+1] = temp1;
                i = j-1;
        }
        y_data[i+1] = scalar;
}


// Define new data-set

for (i = 1; i<=loop; i++)
{
        if (y_data[i] == new_fd) index = i;  // locate index of new/re-odered data-point
        temp1 = y_data[i];
        y_new[i] = temp1;
        x_new[I] = i;
}


// Apply linear fit routine

linear_fit(loop,x_new, y_new, &m, &c);
printf("Value of slope: %f\n\n", m);
printf("Value of Y-intercept: %f\n\n", c);

}
```

```
// Least sqaure fit routine

linear_fit(loop,x_data,y_data, m,c)
{


int n, i;
float x[200], y[200], squared[200],mean_x, mean_y, sum_x, sum_y;
float sum_sqx, product_xy, xy[200], temp1, temp2;

sum_x = 0;
sum_y = 0;
sum_sqx = 0;
product_xy = 0;
n = loop;

for (i=1; i<=n; i++)
{

temp1 = x_data[i];
x[i] = temp1;
temp1 = y_data[i];
y[i] = temp1;
printf("x[%d] = %f\n",i, x[i]);
printf("y[%d] = %f\n",i, y[i]);

}

temp1 = 0;


for (i=1; i<= n; i++)
{


sum_x = sum_x + x[i];
sum_y = sum_y + y[i];
temp1 = x[i]*x[i];
squared[i] = temp1;
xy[i] = x[i]*y[i];
sum_sqx = sum_sqx + squared[i];
product_xy = product_xy + xy[i];


}


mean_x = division(sum_x, n);
```

```
mean_y = division(sum_y, n);


// Compute Slope


temp1 = ((n*product_xy) - (sum_x*sum_y));
temp2 = (n*sum_sqx) - (sum_x*sum_x);

m = division(temp1,temp2);

// Compute Y-intercept

c = mean_y - (m*mean_x);



}
```

PROGRAM LISTINGS 4: GENERAL INTERFEROGRAM PROCESSING AND PREPROCESSING ALGORITHMS

```
/***** FILENAME: "C:\VISION_P\CUTED1E.RSC" *****/

/****
        This program is used to filter noisy frequencies using a FFT
        method


         Written by:  Ade Idowu
         Date: 29/8/96
         Version 2.0
                                                                    ***/


cutt_example()


{
// Open and display image

       init.IWINDOW = 0;
       PopUpImageWindow();


         read(
       /* Input file */       "c:/vision_p/error/n1_360"
       /* Output image */     ,"input"
       /* Type */     ,0
       );

// Define an area of interest (AOI)
cut("input", {60,80}, {128,128}, "output");
normalize("output",0, {0,255}, {0,160}, "drain");
subtract("drain", "drain", "tem");
cut("drain", {50,15}, {58,90}, "drain_cut");
paste("drain_cut", "tem",{38,19});
fft2d("tem", "re", "img");
xytortheta("re", "img", "mod", "ph");
center("mod", "modul");
center("ph", "phcen");

// Subtract specified frequencies
```

```
subtract("mod", "mod", "d1");
subtract("ph", "ph", "d2");

// Extract the the frequencies
cut("modul", {57,59}, {16,16}, "modcut");
cut("phcen", {57,59}, {16,16}, "phcut");

// Paste the frequency
paste("modcut", "d1", {57,59});
paste("phcut", "d2", {57,59});

// Subtract specified frequencies
subtract("modul", "d1", "result1");
subtract("phcen", "d2", "result2");

// Centre the FFT image
center("result1", "mod1");
center("result2", "ph1");

// Inverse FFT
rthetatoxy("mod1", "ph1", "re1", "img");
fft2dinv("re1", "img", "out_re1", "out_img");
// cut("out", {30,30}, {8,8}, "out2");

center("d1", "d1_result");
center("d2", "d2_result");

rthetatoxy("d1_result", "d2_result", "d1_r", "d2_r");
fft2dinv("d1_r", "d2_r", "out_re2", "out_img2");

}
```

```
#include "d:/vision_p/time.h"
#include "d:/vision_p/resolve5.h"
#define LOOP 60
#define ANGLE_COUNT 4000



/****

        PROGRAM: 'FRINGE_ARQUIRE
        DEVELOPED BY : ADE IDOWU
        VERSION: 2.0 (22/10/96)
NOTE: This program samples the fringe pattern at specifed
angular intervals of n degress.

                                                ******/


resolver_ac()

{


    int j,i,angular_value,temp;
    char result[200], a[200], b[200];


for (j=1; j<=LOOP; j++)
{

  a[j] = "d:/vision_p/images/c2n_";
  temp = j*(360/LOOP);
  b[j] =  temp;


  strncat(a[j], b[j], 1000000);
  result[j] = a[j];

 }


for (j=1; j<=LOOP; j++)
{
  for (i=1; i<=ANGLE_COUNT; i++)
{
    temp = j*(360/LOOP);
  angular_value = abs(resolver());


   if(angular_value ==temp) {
```

```
  snap("temp", 1);
  break;   }

        }

save("temp", result[j],3 );


}

  temp = LOOP;
  return temp;

}
```

```
// FILENAME: C:/VISION_P/STATIS2.RSC

/**

        PROGRAM DESCRIPTION: This program is used to measure relevant statistics
        of interferograms

    DEVELOPED BY: ADE IDOWU  (3/7/96)
    Version 4.1

                        **/

#include "d:/vision_p/store1.rsc"
#include "d:/vision_p/c5_servo.rsc"
#define LEFT   48
#define RIGHT 32
#define UP  128
#define DOWN  192


 interferogram_statistics()

 {

// Definition of program variables

int on_time = 1;
int on_time_interval = 14;
int i, j;
float mean, maximum, volume;
char a[20],b[20],c[20],result[20];



   init.IWINDOW=0;
   PopUpImageWindow();


  for (j=1; j <=10; j++)  {

     for(i=1; i<=on_time_interval; i++)    {


//  START OFF BY CAPTURING THE PATTERN, SPECIFY AN A.O.I AND MODIFY ITS
CONTRAST

      snap("ImgName", "1");
      copy("ImgName", "input");
```

```
    cut("input", {60,80}, {128,128}, "output");
    normalize("output",0, {0,255}, {0,160}, "drain");

// CALCULATE THE RELEVANT STATISTICS OF THE INTERFEROGRAM

   extrema("drain");
   mean = MEAN;
   volume = VOLUME;
   maximum = MAXIMUM;


// NOW A ONE-SECOND SERVO ACTUATION IS APPLIED TO PROVIDE A NEW
INTERFEROGRAM

        timming(RIGHT,on_time);


// REPEAT THE STORAGE ROUTINE AGAIN

    a[j] = "d:/vision_p/store1";
    b[j] = j;
    c[j] = ".txt";


 strncat(a[j], b[j], 1000000);

 strncat(a[j], c[j], 100000);

 result[j] = a[j];



    storage1(result[j], i, mean, volume, maximum);

                                        }


            timming(LEFT,16);
                                        }
}
```

PROGRAM LISTINGS 5: PROGRAM USED TO CALCULATING THE SPINDLE 3 DOF ERROR MOTIONS

```
% FILENAME: AXIAL.M
% This program is used to calculate spindle AXIAL error motions
%
%
% DEVELOPED BY: ADE IDOWU
% DATE: 2/3/97
% Version 6.0



% Open output file to read its data
fid = fopen('c:\vision_p\data\out2_1.txt', 'r');
%fid = fopen(filename, 'r');


% Start to sample data from the beginning of loop
for i = 1:31

% Read FRINGE PEAK POSITION (x)
x1(i) = fscanf(fid,'%g',1);


% Read FRINGE DENSITY (p)
p1(i) = fscanf(fid,'%g',1);

% Read FRINGE SPACING (s)
s1(i) = fscanf(fid,'%g',1);

% Read FRINGE ORIENTATION (o)
o1(i) = fscanf(fid,'%g',1);


% End sampling of data
end
end

% Close the output file
fclose(fid);


% Open output file to read its data
fid = fopen('c:\vision_p\data\out2_2.txt', 'r');
%fid = fopen(filename, 'r');
```

```
% Start to sample data from the beginning of loop
for i = 1:31


% Read FRINGE PEAK POSITION (x)
x2(i) = fscanf(fid,'%g',1);


% Read FRINGE DENSITY (p)
p2(i) = fscanf(fid,'%g',1);

% Read FRINGE SPACING (s)
s2(i) = fscanf(fid,'%g',1);

% Read FRINGE ORIENTATION (o)
o2(i) = fscanf(fid,'%g',1);


% End sampling of data
end
end

% Close the output file
fclose(fid);


% Open output file to read its data
fid = fopen('c:\vision_p\data\out2_3.txt', 'r');
%fid = fopen(filename, 'r');


% Start to sample data from the beginning of loop
for i = 1:31


% Read FRINGE PEAK POSITION (x)
x3(i) = fscanf(fid,'%g',1);


% Read FRINGE DENSITY (p)
p3(i) = fscanf(fid,'%g',1);

% Read FRINGE SPACING (s)
s3(i) = fscanf(fid,'%g',1);

% Read FRINGE ORIENTATION (o)
o3(i) = fscanf(fid,'%g',1);


% End sampling of data
end
end
```

```
% Close the output file
fclose(fid);


% Open output file to read its data
fid = fopen('c:\vision_p\data\out2_4.txt', 'r');
%fid = fopen(filename, 'r');


% Start to sample data from the beginning of loop
for i = 1:31

% Read FRINGE PEAK POSITION (x)
x4(i) = fscanf(fid,'%g',1);


% Read FRINGE DENSITY (p)
p4(i) = fscanf(fid,'%g',1);

% Read FRINGE SPACING (s)
s4(i) = fscanf(fid,'%g',1);

% Read FRINGE ORIENTATION (o)
o4(i) = fscanf(fid,'%g',1);


% End sampling of data
end
end

% Close the output file
fclose(fid);


% Open output file to read its data
fid = fopen('c:\vision_p\data\out2_5.txt', 'r');
%fid = fopen(filename, 'r');


% Start to sample data from the beginning of loop
for i = 1:31


% Read FRINGE PEAK POSITION (x)
x5(i) = fscanf(fid,'%g',1);


% Read FRINGE DENSITY (p)
p5(i) = fscanf(fid,'%g',1);

% Read FRINGE SPACING (s)
s5(i) = fscanf(fid,'%g',1);
```

```
% Read FRINGE ORIENTATION (o)
o5(i) = fscanf(fid,'%g',1);



% End sampling of data
end
end

% Close the output file
fclose(fid);

% Calculate the spindle angle in radians:
k = 0:12:360;
angle = k*pi/180;
Fs = 31;
samples = (0:(Fs-1))/(Fs-1);


% Calculate axial error:

% Calculate the reference values
for i = 1:Fs
xr(i) = ((x1(i) + x2(i) + x3(i) + x4(i) + x5(i)))/5;
sr(i) = ((s1(i) + s2(i) + s3(i) + s4(i) + s5(i)))/5;
end

[y,n] = min(x1);
n = 30;
z=mean(xr);
s_mean = mean(sr);
for i = 1:Fs
a1(i) = x1(n) - x1(i);
sr1(i)= s1(n) - s1(i);
a1(i) = a1(i) - sr1(i);

a2(i) = x1(n) - x2(i);
sr2(i)= s1(n) - s2(i);
a2(i) = a2(i) - sr2(i);

a3(i) = x1(n) - x3(i);
sr3(i)= s1(n) - s3(i);
a3(i) = a3(i) - sr3(i);

a4(i) = x1(n) - x4(i);
sr4(i) =s1(n) - s4(i);
a4(i) = a4(i) - sr4(i);

a5(i) = x1(n) - x5(i);
sr5(i)= s1(n) - s5(i);
a5(i) = a5(i) - sr5(i);
end
```

```
for i = 1:Fs
position_av(i) = (a1(i) + a2(i) + a3(i) + a4(i) + a5(i))/5;
asyn1(i) = a1(i) - position_av(i);
asyn2(i) = a2(i) - position_av(i);
asyn3(i) = a3(i) - position_av(i);
asyn4(i) = a4(i) - position_av(i);
asyn5(i) = a5(i) - position_av(i);
av_asyn(i) = (asyn1(i)+asyn2(i)+asyn3(i)+asyn4(i)+asyn5(i))/5;
p_av(i) = (p1(i) + p2(i) + p3(i) + p4(i) + p5(i))/5;
mean_o(i) = (o1(i) + o2(i) + o3(i) + o4(i) + o5(i))/5;
mean_o(i) = mean_o(i) * pi/180;
end

for i = 1:Fs
xav(i)= (position_av(i))*cos(mean_o(i));
yav(i) =  (position_av(i))*sin(mean_o(i)) -  (position_av(i))*0 ; %
magav(i) = sqrtm((xav(i).^2) + (yav(i).^2));
% get the sign
 magav(i) = magav(i) * (position_av(i)/abs(position_av(i)));
end
i = 1 : Fs;
plot(i, magav);
pause


% Calculate the asychronous axial errors
for i = 1:Fs
axial1(i) = a1(i)/x1(i) * 633;
axial2(i) = a2(i)/x2(i) * 633;
axial3(i) = a3(i)/x3(i) * 633;
axial4(i) = a4(i)/x4(i) * 633;
axial5(i) = a5(i)/x5(i) * 633;
end

% Polar display correction
axial1(1) = axial1(31);
axial2(1) = axial2(31);
axial3(1) = axial3(31);
axial4(1) = axial4(31);
axial5(1) = axial5(31);

%Plot asynchronous 'pitch' polar chart using eqaution in thesis:
for i = 1:Fs
h1(i) = (axial1(i)+2*max(axial1))*cos(angle(i));
v1(i) = (axial1(i)+2*max(axial1))*sin(angle(i));
h2(i) = (axial2(i)+2*max(axial1))*cos(angle(i));
v2(i) = (axial2(i)+2*max(axial1))*sin(angle(i));
h3(i) = (axial3(i)+2*max(axial1))*cos(angle(i));
v3(i) = (axial3(i)+2*max(axial1))*sin(angle(i));
h4(i) = (axial4(i)+2*max(axial1))*cos(angle(i));
v4(i) = (axial4(i)+2*max(axial1))*sin(angle(i));
h5(i) = (axial5(i)+2*max(axial1))*cos(angle(i));
v5(i) = (axial5(i)+2*max(axial1))*sin(angle(i));
end
```

```
% Evaluate asynchronous (axial) error motion (as):
min_e(1) = min(axial1);max_e(1)= max(axial1);
min_e(2) = min(axial2);max_e(2)= max(axial2);
min_e(3) = min(axial3);max_e(3)= max(axial3);
min_e(4) = min(axial4);max_e(4)= max(axial4);
min_e(5) = min(axial5);max_e(5)= max(axial5);
es = abs(min(min_e)) + max(max_e);

% Re-plot polar chart of spindle 'pitch' error
% stating the error motion value in urads:
plot(h1,v1,'m',h2,v2,'c',h3,v4,'r',h4,v4,'g',h5,v5,'b')
xlabel('Displacement (nm)')
ylabel('Displacement (nm)')
title(' Unfiltered total error of spindle: 520 nm ')
pause


% Calculate the asychronous axial errors
for i = 1:Fs
av_axial(i) = magav(i)/x1(i) * 633;
h(i) = (av_axial(i)+max(av_axial))*cos(angle(i));
v(i) = (av_axial(i)+max(av_axial))*sin(angle(i));
end

% Polar display correction
h(1) = h(31);

% Evaluate asynchronous (axial) error motion (as):
min_axial(1) = min(av_axial);max_axial(1)= max(av_axial);
axial_s = abs(min(min_axial)) + max(max_axial);


% Re-plot polar chart of spindle axial error
% stating the error motion value in urads:
plot(h,v,'m')
xlabel('Displacement (nm)')
ylabel('Displacement (nm)')
title(' Unfiltered average axial error of spindle: 261 nm ')
pause


% Frequency spectra of data

%[b,a] = ellip(4,2,30,[0.040 0.055]/0.066);
%[b,a] = ellip(4,0.10,30,[0.025 0.037]/0.066);
[b,a] = ellip(8,0.30,30,[0.030 0.045]/0.066);

[H,w] =freqz(b,a,512);
w = (0:255)/(256*(Fs/2));
sfa =filter(b,a,av_axial);
SA= fft(av_axial,512);
SFA = fft(sfa,512);
sf1 =filter(b,a,axial1);
S1= fft(axial1,512);
```

```
SF1 = fft(sf1,512);
sf2 =filter(b,a,axial2);
S2= fft(axial2,512);
SF2 = fft(sf2,512);
sf3 =filter(b,a,axial3);
S3= fft(axial3,512);
SF3 = fft(sf3,512);
sf4 =filter(b,a,axial4);
S4= fft(axial4,512);
SF4 = fft(sf4,512);
sf5 =filter(b,a,axial5);
S5= fft(axial5,512);
SF5 = fft(sf5,512);

%Plot elliptic filter characteristics
w = w*15.6212;
plot(w,abs(SA(1:256)),'b-',w,abs(SFA(1:256)), 'r-')
xlabel(' Rotational Frequency (Hz) ');
ylabel(' Amplitude (nm) ' )
pause

%Plot elliptic filter characteristics
freqz(b,a);
pause

%Plot (filtered) synchronous 'axial' polar chart usingeqaution in thesis:
for i = 1:Fs
hm(i) = (sfa(i)+3*max(sfa))*cos(angle(i));
vm(i) = (sfa(i)+3*max(sfa))*sin(angle(i));
end

% Evaluate (filtered) synchronous (pitch) error motion (ep):
min_sf = min(sfa);max_sf= max(sfa);
epf = abs(min_sf) + max_sf;

% Re-plot polar chart of spindle 'axial' error
% stating the error motion value in nm:
hm(1) = hm(31);
plot(hm,vm, 'b-');
xlabel('Displacement (nm)')
ylabel('Displacement (nm)')
title(' Filtered average axial error: 82.2 nm')
pause

%Plot elliptic filter characteristics
plot(w,abs(S4(1:256)),'b-',w,abs(SF4(1:256)), 'r-')
xlabel(' Frequency (sampling rate) in Hz ');
ylabel(' Power of signal ' )
```

```
% FILENAME: TILT.M
% This program is used to calculate spindle TILT error motions
%
%
%
% DEVELOPED BY: ADE IDOWU
% DATE: 2/6/97
% Version 6.0



% Open output file to read its data
fid = fopen('c:\vision_p\data\dat_1.txt', 'r');
%fid = fopen(filename, 'r');


% Start to sample data from the beginning of loop
for i = 1:31


% Read FRINGE DENSITY (p)
p1(i) = fscanf(fid,'%g',1);
value = p1(i);

% Read SPINDLE POSITION (m)
m1(i) = fscanf(fid,'%g',1);

% Read FRINGE ORIENTATION (o)
o1(i) = fscanf(fid,'%g',1);

% Read FRINGE SPATIAL FREQUENCY (f)
f1(i) = fscanf(fid,'%g',1);

% Read X COORDINATE OF FIRST PEAK(x_p)
x_p1(i) = fscanf(fid,'%g',1);

% Read Y COORDINATE OF FIRST PEAK(y_p)
y_p1(i) = fscanf(fid,'%g',1);


% Read FRINGE POSITION (x)
for j = 1:value
temp(j) = fscanf(fid,'%g',1);
x1(i,j) = temp(j);


% End sampling of data
end
end

% Close the output file
fclose(fid);


% Open output file to read its data
```

```
fid = fopen('c:\vision_p\data\dat_2.txt', 'r');
%fid = fopen(filename, 'r');


% Start to sample data from the beginning of loop
for i = 1:31


% Read FRINGE DENSITY (p)
p2(i) = fscanf(fid,'%g',1);
value = p2(i);

% Read SPINDLE POSITION (m)
m2(i) = fscanf(fid,'%g',1);

% Read FRINGE ORIENTATION (o)
o2(i) = fscanf(fid,'%g',1);

% Read FRINGE SPATIAL FREQUENCY (f)
f2(i) = fscanf(fid,'%g',1);

% Read X COORDINATE OF FIRST PEAK(x_p)
x_p2(i) = fscanf(fid,'%g',1);

% Read Y COORDINATE OF FIRST PEAK(y_p)
y_p2(i) = fscanf(fid,'%g',1);


% Read FRINGE POSITION (x)
for j = 1:value
temp(j) = fscanf(fid,'%g',1);
x2(i,j) = temp(j);


% End sampling of data
end
end

% Close the output file
fclose(fid);

% Open output file to read its data
fid = fopen('c:\vision_p\data\dat_3.txt', 'r');
%fid = fopen(filename, 'r');


% Start to sample data from the beginning of loop
for i = 1:31


% Read FRINGE DENSITY (p)
p3(i) = fscanf(fid,'%g',1);
value = p3(i);

% Read SPINDLE POSITION (m)
```

```
m3(i) = fscanf(fid,'%g',1);

% Read FRINGE ORIENTATION (o)
o3(i) = fscanf(fid,'%g',1);

% Read FRINGE SPATIAL FREQUENCY (f)
f3(i) = fscanf(fid,'%g',1);

% Read X COORDINATE OF FIRST PEAK(x_p)
x_p3(i) = fscanf(fid,'%g',1);

% Read Y COORDINATE OF FIRST PEAK(y_p)
y_p3(i) = fscanf(fid,'%g',1);


% Read FRINGE POSITION (x)
for j = 1:value
temp(j) = fscanf(fid,'%g',1);
x3(i,j) = temp(j);


% End sampling of data
end
end

% Close the output file
fclose(fid);


% Open output file to read its data
fid = fopen('c:\vision_p\data\dat_4.txt', 'r');
%fid = fopen(filename, 'r');


% Start to sample data from the beginning of loop
for i = 1:31


% Read FRINGE DENSITY (p)
p4(i) = fscanf(fid,'%g',1);
value = p4(i);

% Read SPINDLE POSITION (m)
m4(i) = fscanf(fid,'%g',1);

% Read FRINGE ORIENTATION (o)
o4(i) = fscanf(fid,'%g',1);

% Read FRINGE SPATIAL FREQUENCY (f)
f4(i) = fscanf(fid,'%g',1);

% Read X COORDINATE OF FIRST PEAK(x_p)
x_p4(i) = fscanf(fid,'%g',1);

% Read Y COORDINATE OF FIRST PEAK(y_p)
```

```
y_p4(i) = fscanf(fid,'%g',1);


% Read FRINGE POSITION (x)
for j = 1:value
temp(j) = fscanf(fid,'%g',1);
x4(i,j) = temp(j);


% End sampling of data
end
end
```

```
% Close the output file
fclose(fid);


% Open output file to read its data
fid = fopen('c:\vision_p\data\dat_5.txt', 'r');
%fid = fopen(filename, 'r');


% Start to sample data from the beginning of loop
for i = 1:31


% Read FRINGE DENSITY (p)
p5(i) = fscanf(fid,'%g',1);
value = p5(i);
```

```
% Read SPINDLE POSITION (m)
m5(i) = fscanf(fid,'%g',1);

% Read FRINGE ORIENTATION (o)
o5(i) = fscanf(fid,'%g',1);

% Read FRINGE SPATIAL FREQUENCY (f)
f5(i) = fscanf(fid,'%g',1);

% Read X COORDINATE OF FIRST PEAK(x_p)
x_p5(i) = fscanf(fid,'%g',1);

% Read Y COORDINATE OF FIRST PEAK(y_p)
y_p5(i) = fscanf(fid,'%g',1);

% Read FRINGE POSITION (x)
for j = 1:value
temp(j) = fscanf(fid,'%g',1);
x5(i,j) = temp(j);


% End sampling of data
end
end
```

```
% Close the output file
fclose(fid);


% Convert orientation angles from radians to degress:
o1 = o1*pi/180;
o2 = o2*pi/180;
o3 = o3*pi/180;
o4 = o4*pi/180;
o5 = o5*pi/180;


% Find mean values:
mean_o = (o1 + o2 + o3 + o4 + o5)/5;
mean_f = (f1 + f2 + f3 + f4 + f5)/5;


% Calculate reference datums
m=15;
o_ref = o1(m);
f_ref = f1(m);


% Calculate the spindle angle in radians:
k = 0:12:360;
angle = k*pi/180;
Fs = 31;
samples = (0:(Fs-1))/(Fs-1);


% Calculate angle differences
for i = 1:Fs
delta_o1(i) = o1(i)-mean_o(i); delta_o1(i) = (cos(delta_o1(i)));
delta_o2(i) = o2(i)-mean_o(i); delta_o2(i) = (cos(delta_o2(i)));
delta_o3(i) = o3(i)-mean_o(i); delta_o3(i) = (cos(delta_o3(i)));
delta_o4(i) = o4(i)-mean_o(i); delta_o4(i) = (cos(delta_o4(i)));
delta_o5(i) = o5(i)-mean_o(i); delta_o5(i) = (cos(delta_o5(i)));
end

% Calculate difference in spatial frequency due to spindle tilt
for i = 1:Fs
delta_f1(i) = ((f_ref)/delta_o1(i))-f1(i);
delta_f2(i) = ((f_ref)/delta_o2(i))-f2(i);
delta_f3(i) = ((f_ref)/delta_o3(i))-f3(i);
delta_f4(i) = ((f_ref)/delta_o4(i))-f4(i);
delta_f5(i) = ((f_ref)/delta_o5(i))-f5(i);


% Calculate the tilt vector value:
tilt_1(i) = (delta_f1(i))/(f1(i)) * 635e-9;
tilt_2(i) = (delta_f2(i))/(f2(i)) * 635e-9;
tilt_3(i) = (delta_f3(i))/(f3(i)) * 635e-9;
tilt_4(i) = (delta_f4(i))/(f4(i)) * 635e-9;
tilt_5(i) = (delta_f5(i))/(f3(i)) * 635e-9;
```

```
% Calculate the othogonal-X tilt vector value:
xtilt_1(i) = tilt_1(i)*cos(o1(i));
xtilt_2(i) = tilt_2(i)*cos(o2(i));
xtilt_3(i) = tilt_3(i)*cos(o3(i));
xtilt_4(i) = tilt_4(i)*cos(o4(i));
xtilt_5(i) = tilt_5(i)*cos(o5(i));

% Calculate the othogonal-Y tilt vector value:
ytilt_1(i) = tilt_1(i)*sin(o1(i));
ytilt_2(i) = tilt_2(i)*sin(o2(i));
ytilt_3(i) = tilt_3(i)*sin(o3(i));
ytilt_4(i) = tilt_4(i)*sin(o4(i));
ytilt_5(i) = tilt_5(i)*sin(o5(i));

% Calculate the tilt (angular) error :
angle_1(i) =atan(tilt_1(i)/0.1);
angle_2(i) =atan(tilt_2(i)/0.1);
angle_3(i) =atan(tilt_3(i)/0.1);
angle_4(i) =atan(tilt_4(i)/0.1);
angle_5(i) =atan(tilt_5(i)/0.1);


% Calculate the othogonal-X tilt (angular) error :
xangle_1(i) =atan(xtilt_1(i)/0.1);
xangle_2(i) =atan(xtilt_2(i)/0.1);
xangle_3(i) =atan(xtilt_3(i)/0.1);
xangle_4(i) =atan(xtilt_4(i)/0.1);
xangle_5(i) =atan(xtilt_5(i)/0.1);

% Calculate the othogonal-Y tilt (angular) error :
yangle_1(i) =atan(ytilt_1(i)/0.1);
yangle_2(i) =atan(ytilt_2(i)/0.1);
yangle_3(i) =atan(ytilt_3(i)/0.1);
yangle_4(i) =atan(ytilt_4(i)/0.1);
yangle_5(i) =atan(ytilt_5(i)/0.1);

end

% Calculate the mean value of the spatial frequency difference
mean_f =(delta_f1 + delta_f2 + delta_f3 + delta_f4 + delta_f5)/5;

% Calculate the mean value of the x_tilt (angular) error:
mean_tx =(xangle_1 + xangle_2 + xangle_3 + xangle_4 +xangle_5)/5;

% Calculate the mean value of the y_tilt (angular) error:
mean_ty =(yangle_1 + yangle_2 + yangle_3 + yangle_4 +yangle_5)/5;

%Plot polar chart of Pitch error motion:
subplot(2,3,1)
polar(angle,max(2*xangle_1)+xangle_1,'r-');
subplot(2,3,2)
polar(angle,max(2*xangle_2)+xangle_2,'r-');
subplot(2,3,3)
polar(angle,max(2*xangle_3)+xangle_3,'r-');
```

```
subplot(2,3,4)
polar(angle,max(2*xangle_4)+xangle_4,'r-');
subplot(2,3,5)
polar(angle,max(2*xangle_5)+xangle_5,'r-');
xlabel('                Pitch errors in urads');
title('Polar plots for five samples of data')
pause
clf reset


%Plot asynchronous 'pitch' polar chart using eqaution in thesis:
for i = 1:Fs
h1(i) = (xangle_1(i)+5*max(xangle_1))*cos(angle(i));
v1(i) = (xangle_1(i)+5*max(xangle_1))*sin(angle(i));
h2(i) = (xangle_2(i)+5*max(xangle_1))*cos(angle(i));
v2(i) = (xangle_2(i)+5*max(xangle_1))*sin(angle(i));
h3(i) = (xangle_3(i)+5*max(xangle_1))*cos(angle(i));
v3(i) = (xangle_3(i)+5*max(xangle_1))*sin(angle(i));
h4(i) = (xangle_4(i)+5*max(xangle_1))*cos(angle(i));
v4(i) = (xangle_4(i)+5*max(xangle_1))*sin(angle(i));
h5(i) = (xangle_5(i)+5*max(xangle_1))*cos(angle(i));
v5(i) = (xangle_5(i)+5*max(xangle_1))*sin(angle(i));
end


% Evaluate asychronous (pitch) error motion (ep):
min_e(1) = min(xangle_1);max_e(1)= max(xangle_1);
min_e(2) = min(xangle_2);max_e(2)= max(xangle_2);
min_e(3) = min(xangle_3);max_e(3)= max(xangle_3);
min_e(4) = min(xangle_4);max_e(4)= max(xangle_4);
min_e(5) = min(xangle_5);max_e(5)= max(xangle_5);
aep = abs(min(min_e)) + max(max_e);

% Re-plot polar chart of spindle 'pitch' error
% stating the error motion value in urads:
plot(h1,v1,'m',h2,v2,'c',h3,v4,'r',h4,v4,'g',h5,v5,'b')
xlabel('rads')
ylabel('rads')
title('      Asychronous pitch error: 1.9 urads');
%text(-1.5e-6, 3e-6,' Asychronous tilt (pitch) error of spindle: 1.24 urads')
pause


%Plot asynchronous 'roll' polar chart using eqaution in thesis:
for i = 1:Fs
h1(i) = (yangle_1(i)+max(yangle_1))*cos(angle(i));
v1(i) = (yangle_1(i)+max(yangle_1))*sin(angle(i));
h2(i) = (yangle_2(i)+max(yangle_1))*cos(angle(i));
v2(i) = (yangle_2(i)+max(yangle_1))*sin(angle(i));
h3(i) = (yangle_3(i)+max(yangle_1))*cos(angle(i));
v3(i) = (yangle_3(i)+max(yangle_1))*sin(angle(i));
h4(i) = (yangle_4(i)+max(yangle_1))*cos(angle(i));
v4(i) = (yangle_4(i)+max(yangle_1))*sin(angle(i));
h5(i) = (yangle_5(i)+max(yangle_1))*cos(angle(i));
v5(i) = (yangle_5(i)+max(yangle_1))*sin(angle(i));
```

```
end

% Evaluate asynchronous (roll) error motion (et):
min_e(1) = min(yangle_1);max_e(1)= max(yangle_1);
min_e(2) = min(yangle_2);max_e(2)= max(yangle_2);
min_e(3) = min(yangle_3);max_e(3)= max(yangle_3);
min_e(4) = min(yangle_4);max_e(4)= max(yangle_4);
min_e(5) = min(yangle_5);max_e(5)= max(yangle_5);
aet = abs(min(min_e)) + max(max_e);

% Re-plot polar chart of spindle 'roll' error
% stating the error motion value in urads:
plot(h1,v1,'m',h2,v2,'c',h3,v4,'r',h4,v4,'g',h5,v5,'b')
xlabel('rads')
ylabel('rads')
title('     Asychronous roll error: 1.9 urads')
pause


%Plot (unfiltered) average 'pitch' polar chart using eqaution in thesis:
for i = 1:Fs
hm(i) = (mean_tx(i)+2*max(mean_tx))*cos(angle(i));
vm(i) = (mean_tx(i)+2*max(mean_tx))*sin(angle(i));
end

% Evaluate synchronous (pitch) error motion (ep):
min_em = min(mean_tx);max_em= max(mean_tx);
sep = abs(min_em) + max_em;

% Re-plot polar chart of spindle 'roll' error
% stating the error motion value in urads:
plot(hm,vm,'b-' );
xlabel('rads')
ylabel('rads')
title('      Unfiltered average pitch error:1 urad')
pause


%Plot (unfiltered) average 'roll' polar chart using eqaution in thesis:
for i = 1:Fs
hm(i) = (mean_ty(i)+0.5*max(mean_ty))*cos(angle(i));
vm(i) = (mean_ty(i)+0.5*max(mean_ty))*sin(angle(i));
end

% Evaluate synchronous (roll) error motion (et):
min_em = min(mean_ty);max_em= max(mean_ty);
set = abs(min_em) + max_em;

% Re-plot polar chart of spindle 'roll' error
% stating the error motion value in urads:
plot(hm,vm,'b-');
xlabel('rads')
ylabel('rads')
title('      Unfiltered average roll error:1.1 urads')
pause
```

```
clf reset

% Frequency spectra of data
%[b,a] = ellip(4,2,40,[0.001 0.012]/0.066);
[b,a] = ellip(4,0.10,30,[0.025 0.037]/0.066);
[d,c] = ellip(4,0.10,30,[0.035 0.042]/0.066);
[H,w] =freqz(b,a,512);
w = (0:255)/(256*(Fs/2));
sf1 =filter(b,a,mean_tx);
sf2 =filter(d,c,mean_ty);
S1= fft(mean_tx,512);
S2 = fft(mean_ty,512);
SF1 = fft(sf1,512);
SF2 = fft(sf2,512);

%Plot elliptic filter characteristics
freqz(b,a);
pause
w = w*15.6212; % Frequency is measured as number of data points per cycle
plot(w,abs(S1(1:256)),'b-',w,abs(SF1(1:256)), 'r-')
xlabel(' Rotational Frequency (Hz) ');
ylabel(' Amplitude (rads) ' )
pause
plot(w,abs(S2(1:256)),'b-',w,abs(SF2(1:256)), 'r-')
xlabel(' Frequency (Rotational Frequency (Hz)');
ylabel(' Amplitude (rads) ' )
pause

%Plot (filtered) synchronous 'pitch' polar chart using eqaution in thesis:
for i = 1:Fs
hm(i) = (sf1(i)+5*max(sf1))*cos(angle(i));
vm(i) = (sf1(i)+5*max(sf1))*sin(angle(i));
end

% Evaluate (filtered) synchronous (pitch) error motion (ep):
min_sf1 = min(sf1);max_sf1= max(sf1);
epf = abs(min_sf1) + max_sf1;

% Re-plot polar chart of spindle 'roll' error
% stating the error motion value in urads:
hm(1) = hm(31);
plot(hm,vm, 'b-');
xlabel('rads')
ylabel('rads')
title('      Filtered average pitch error: 0.19 urad');
pause

%Plot (filtered) synchronous 'roll' polar chart using eqaution in thesis:
for i = 1:Fs
hm(i) = (sf2(i)+5*max(sf2))*cos(angle(i));
vm(i) = (sf2(i)+5*max(sf2))*sin(angle(i));
end
```

```
% Evaluate (filtered) synchronous (roll) error motion (et):
min_sf2 = min(sf2);max_sf2 = max(sf2);
etf = abs(min_sf2) + max_sf2;

% Re-plot polar chart of spindle 'roll' error
% stating the error motion value in urads:
hm(1) = hm(31);
plot(hm,vm,'b-');
xlabel('rads')
ylabel('rads')
title('       Filtered average roll error: 0.12 urad');
```

PROGRAM LISTINGS 6: PROGRAMS USED TO ANALYSE INFERFEROGRAM INTENSITY PROFILES AND APPLY LEAST SQUARE POLYNOMIAL FITS

```
% PROGRAM DESCRIPTION: This program is used to read data created by the interferogram
% software. It is used to read/ analyse the interferogram pixel data, and to sub-pixel
% interpolation.
%
% DEVELOPED BY: ADE IDOWU
%DATE: (2/7/98)
% Version 3.0
%

%function read_1(filename)

clear

for file_num = 1:5



        string1 = 'c:/vision_p/mat_lab/fringe_ .txt';
        string2 = ' ';
        string3 = num2str(file_num);
        filename = strrep(string1, string2, string3);
        fid = fopen(filename, 'r');

% Start to sample data from the beginning of loop
for i = 1:30
        % Read FRINGE DENSITY
        fringe_density(i,file_num) = fscanf(fid,'%g',1);

end

% Close the output file
fclose(fid);


        string1 = 'c:/vision_p/mat_lab/flag_ .txt';
        filename =strrep(string1, string2, string3);
        fid = fopen(filename, 'r');

% Start to sample data from the beginning of loop
for i = 1:30
        value_1(i,file_num) = fscanf(fid,'%g',1);
```

```
        value_2(i,file_num) = fscanf(fid,'%g',1);
end

% Close the output file
fclose(fid);



string1 = 'c:/vision_p/mat_lab/info_ .txt';
filename = strrep(string1, string2, string3);
fid = fopen(filename, 'r');

for i = 1:30
        for j = 1: (value_2(i,file_num) - value_1(i,file_num) + 1)
                left_spacing(i,j) = fscanf(fid,'%g',1);
                right_spacing(i,j) = fscanf(fid,'%g',1);
        end
end

% Close the output file
fclose(fid);



string1 = 'c:/vision_p/mat_lab/data_ .txt';
filename = strrep(string1, string2, string3);
fid = fopen(filename, 'r');


for i = 1:30
for j = 1: (value_2(i,file_num) - value_1(i,file_num)+1)
        for k = left_spacing(i,j) : right_spacing(i,j)
                z(k) = k;
                x(k) = fscanf(fid,'%g',1);
                y(k) = fscanf(fid,'%g',1);
        end
        for order = 1:4;
                start = left_spacing(i,j);
                w = left_spacing(i,j) : right_spacing(i,j);
                value1 = z(w);
                value2 = y(w);
                pos_min = min(value1);
                pos_max = max(value1);
                int_min = min(value2);
                int_max = max(value2);
                a = polyfit(value1,value2,order);
                p = a;
                intensity(order,w)  = polyval(a,value1);
                diff_val = abs(intensity(order,w) - value2);
                error(order) = mean(diff_val);

                d_step = 0.01; %100th of a pixel
                d = pos_min: d_step:pos_max;
                if order==1   c = p(1)*d + p(2); end
                if order==2   c = p(1)*d.^2 + p(2)*d + p(3); end
```

```
                        if order==3   c = p(1)*d.^3+p(2)*d.^2 + p(3)*d + p(4); end
                        if order==4   c = p(1)*d.^4+p(2)*d.^3+p(3)*d.^2 + p(4)*d + p(5); end
                        if order==5   c = p(1)*d.^5+p(2)*d.^4+p(3)*d.^3+p(4)*d.^2 + p(5)*d + p(6);
end
                        if order==6   c = p(1)*d.^6+p(2)*d.^5+p(3)*d.^4+p(4)*d.^3+p(5)*d.^2 +...
                        p(6)*d + p(7); end
                        if order==7   c =
p(1)*d.^7+p(2)*d.^6+p(3)*d.^5+p(4)*d.^4+p(5)*d.^3+p(6)*d.^2 +...
                        p(7)*d + p(8); end
                        if order==8   c =
p(1)*d.^8+p(2)*d.^7+p(3)*d.^6+p(4)*d.^5+p(5)*d.^4+p(6)*d.^3+ ...
                        p(7)*d.^2 + p(8)*d + p(9); end
                        if order==9   c =
p(1)*d.^9+p(2)*d.^8+p(3)*d.^7+p(4)*d.^6+p(5)*d.^5+p(6)*d.^4+ ...
                        p(7)*d.^3+p(8)*d.^2 + p(9)*d + p(10); end
                        if order==10   c =
p(1)*d.^10+p(2)*d.^9+p(3)*d.^8+p(4)*d.^7+p(5)*d.^6+p(6)*d.^5+ ...
                        p(7)*d.^4+p(8)*d.^3+p(9)*d.^2 + p(10)*d + p(11); end

                        h1 = 'At  m = ';
                        h2 = [h1 int2str(order)];
                        %subplot(3,2,order);plot(value1,value2,'b*', d, c, 'r-');
                        %axis([pos_min pos_max int_min int_max+20])
                        %ylabel('Intensity');
                        %text(100,120,h2)
                        %text(100,70,'Spatial position')



end
[min_error index_order] = min(error);
intensity_moder = intensity(index_order,w);

[temp, indexed] = max(intensity_moder);
                        intensity_max(i,j) = temp;
                        position_max(i,j) = indexed+start;
                end
end
% Close the output file
fclose(fid);

if file_num == 1
                intensity_max1 = intensity_max;
                position_max1 = position_max;
elseif file_num == 2
                intensity_max2 = intensity_max;
                position_max2 = position_max;
elseif file_num == 3
                intensity_max3 = intensity_max;
                position_max3 = position_max;
elseif file_num == 4
                intensity_max4 = intensity_max;
                position_max4 = position_max;
```

```
elseif file_num == 5
        intensity_max5 = intensity_max;
        position_max5 = position_max;
end
disp('loop is:');
end

plot(w,intensity_moder,'-b',w,y(left_spacing(i,j) : right_spacing(i,j)) ,'r*')
```

```
% FILENAME: INTEN1.M

% PROGRAM DESCRIPTION: This program is used plot interpolated fringe data for
% the n = 6 order
%
% DEVELOPED BY: ADE IDOWU  (2/7/98)
% Version 3.0
%
%


% Define program variales

b =[ 1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
0
0
0
0
0
0
0
0
0
0
0
0
0
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
```

1
1
1

1
1
1

1
1
1

```
1
1
1
```

147
144
140
137
134
131
128
124
120
117
113
109
106
102
98
95
96
97
98
99
100
102
103
104
105
109
113
117
121
125
129
133
137
141
143
144
146
148
150
152
154
156
158
157
157
157
156
156
155
155
154
152
149
147

144
142
140
137
135
132
128
125
122
119
115
112
109
106
103
104
105
107
108
109
111
112
113
115
117
119
122
124
126
128
130
133
135
137
139
141
143
145
147
148
150
152
152
152
153
153
153
153
153
153
153
153
151
149
147
145

143
141
139
137
135
131
125
121
116
112
107
103
98
94
94
93
93
93
92
91
90
90
90
93
97
101
105
108
112
116
120
124
126
129
131
134
136
138
140
143
145
146
147
148
149
150
151
152
153
154
154
153
153
152
152

151
151
150
150
147
144
141
138
135
132
129
126
123
119
116
113
109
106
103
98
95
91
93
95
96
98
100
102
103
105
107
109
112
115
118
120
123
125
128
131
132
133
134
135
136
137
138
139
140
140
140
140
140
140
140

```
141
141
141
138
135
131
128
125
122
119
116
113
109
105
101
97
93
89
85
80
76
78
79
80
81
83
84
85
87];

max_g = max(g);
```

```
% Plot graphs

 x = 1:length(b);

subplot(1,2,1);plot(x,g,'b+',x,g,'b-');
   %axis([x1_min x1_max y1_min y1_max])
   xlabel('Spatial position');
   ylabel('Pixel grey-levls');

subplot(1,2,2);plot(x,b,'b+',x,b,'b-');
   xlabel('Spatial position');
   ylabel('Pixel grey-levls');
```

```
% FILENAME: FIT_2.M
%
% PROGRAM DESCRIPTION: This program is used plot interpolated fringe data for
% the n = 1 to n = 6 order
%
% DEVELOPED BY: ADE IDOWU  (2/7/98)
% Version 3.0
%
%
% Define program variales
b =[ 111
  112
  113
  115
  117
  119
  122
  124
  126
  128
  130
  133
  135
  137
  139
  141
  143
  145
  147
  148
  150
  152
  152
  152
  153
  153
  153
  153
  153
  153
  153
  153
  151
  149
  147
  145
  143
  141
  139
  137
  135
  131
```

```
125
121
116
112
107
103
 98
 94
 94
 93
 93
 93
 92
 91
 90
 90
 90];

a = 1:58;
a = a';

  a_min = min(a);
  b_min = min(b);
  a_max = max(a);
  b_max = max(b);

  format long       %set output format to 16 decimal places

 no_plots = 6;

 for  n = 1: no_plots
   g = n
   p = polyfit(a,b,n)

 % Use the return coefficients of the polynomial
 d=a;
 d_step = 0.01;   %10th of a pixel
 d = a_min:d_step:a_max;
  if n==1   c = p(1)*d + p(2); end
  if n==2   c = p(1)*d.^2 + p(2)*d + p(3); end
  if n==3   c = p(1)*d.^3+p(2)*d.^2 + p(3)*d + p(4); end
  if n==4   c = p(1)*d.^4+p(2)*d.^3+p(3)*d.^2 + p(4)*d + p(5); end
  if n==5   c = p(1)*d.^5+p(2)*d.^4+p(3)*d.^3+p(4)*d.^2 + p(5)*d + p(6); end
  if n==6   c = p(1)*d.^6+p(2)*d.^5+p(3)*d.^4+p(4)*d.^3+p(5)*d.^2 + p(6)*d + p(7);end
 %  c = p(1)*d.^7+p(2)*d.^6+p(3)*d.^5+p(4)*d.^4+p(5)*d.^3+p(6)*d.^2 + p(7)*d + p(8);
  [max_intense,index] = max(c)
   max_position = d(index)

 h1 = 'At m = '
 h2 = [h1 int2str(n)]


 % Show a plot of the interferogram data and its curve fit
```

```
subplot(3,2,g);plot(a,b,'b*', d, c, 'b-');
axis([a_min a_max b_min b_max+20])
%xlabel('X');
ylabel('Intensity');
text(33.2,120,h2)
text(34,180,'Spatial position')
%text(33.5,130, 'Maximum intensity after curve fitting = 116.0874');
%text(33.5,120, 'Fringe maximum position after curve fitting = 36.5300');

 end
```

PROGRAM LISTINGS 7: PROGRAMS USED TO ANALYSE THE THEORECTICAL/ EXPERIMENTAL MODEL OF INTERFEROMETER TILT VECTOR

```
% FILENAME: SPIN_DEN.M
% DESCRIPTION: This program is used to plot the theorectical and experimental
% characteristic b/w fringe density and angular position
%
%
% DEVELOPED BY: ADE IDOWU
% DATE: 22/12/97

% VERSION: 2.0



% Definition of variables
interval = 6;
angle_cycle = 360;
spin_angle = 0:interval:angle_cycle;
for i = 1:(angle_cycle/interval)+1
count = i;
 a =3;
 b = 8;
 c = 0.0001;
 a1(i) = (a.^2) + (b.^2);
 b1(i) = 2*a*b;
 p(i) = a1(i) + b1(i)*cos((1.0205*spin_angle(i))+90);
 q(i) = p(i).^0.5;
 end
 p_ex = [9

 10
 10
 10
 10
 10
 10
 10
 10
 10
 8
 9
 8
 8
 8
 8
```

```
8
9
7
8
7
7
6
7
6
6
6
6
5
6
6
6
6
6
6
7
6
7
6
7
7
8
7
7
8
8
9
9
9
9
9
10
10
10
9
10
10
11
11
11
11 ];
p_ex = p_ex';
```

```
% Plot graphs

plot(spin_angle, q,'g-', spin_angle, p_ex, 'b-');
axis([0 360 0 15]);
xlabel('Spindle angular position');
ylabel('Fringe density');
```

```
text(250,11.5, 'Theorectical');
text(250,6.5, 'Experimental');
```

```
% FILENAME: SPAT_FRE.M

% DESCRIPTION: This program is used to plot the theorectical and experimental
% characteristic b/w spatial frequency and angular position
%
%
% DEVELOPED BY: ADE IDOWU
% DATE: 22/12/97
% VERSION: 2.0



% Definition of variables

interval = 6;
angle_cycle = 360;
spin_angle = 0:interval:angle_cycle;
for i = 1:(angle_cycle/interval)+1
count = i;
 a = 3;
 b = 8;
 c = 0.0001;
 a1(i) = (a.^2) + (b.^2);
 b1(i) = 2*a*b;
 p(i) = a1(i) + b1(i)*cos((1.0205*spin_angle(i))+90+6);
 q(i) = (p(i).^0.5)*238.65743*1e-04;
 end
 sf_ex = 1e-4*[2245.340
2245.340
2266.780
2200.680
2293.640
2193.260
2103.070
2176.730
2176.730
2224.440
2159.340
2121.430
2126.080
1997.800
2006.960
2072.120
1910.500
2088.070
1751.270
1860.780
1701.310
```

```
1711.730
1504.170
1570.920
1320.060
1492.590
1285.960
1249.470
1166.230
1133.390
1133.390
1133.390
1303.990
1264.970
1367.920
1363.960
1569.260
1442.640
1484.100
1746.430
1686.560
2000.000
1918.620
2054.330
2001.110
2017.700
2247.230
2203.290
2207.080
2191.480
2247.250
2334.580
2334.580
2358.970
2333.410
2394.740
2385.460
2451.650
2372.400
2402.200
2402.200
 ];
sf_ex = sf_ex';


% Plot graphs

 plot(spin_angle, q,'g-', spin_angle, sf_ex, 'b-');
 %axis([0 360 0 15]);
xlabel('Spindle angular position');
ylabel('Spatial Frequency ');
text(250,1800, 'Theorectical');
text(250,2680, 'Experimental');
```